



TITLE:

通信網管理のための組合せ最適化問題の研究(Dissertation_全文)

AUTHOR(S):

伊藤, 大雄

CITATION:

伊藤, 大雄. 通信網管理のための組合せ最適化問題の研究. 京都大学, 1995, 博士(工学)

ISSUE DATE:

1995-03-23

URL:

<https://doi.org/10.11501/3099758>

RIGHT:

新 制

工

997

京大附図

通 信 網 管 理 の た め の
組 合 せ 最 適 化 問 題 の
研 究

1994 年 12 月

伊 藤 大 雄

通信網管理のための 組合せ最適化問題の 研究

1994年12月

伊藤大雄

謝 辞

本研究遂行の際に、終始暖かくかつ適切な御助言を賜り、そして本論文の取り纏めにあたって、懇切なる御指導及び多大なる御尽力を頂いた、京都大学工学部の茨木俊秀教授に衷心より感謝の意を表します。先生の御存在なくして、本研究はもとより、今日の私が存在しないと申しても過言ではありません。

長年に渡って広い御心と深い洞察力に基づいた暖かな御指導を賜り、本研究をまとめるにあたっても、終始励まして下さった、京都大学工学部の長谷川利治教授に、心より感謝申し上げます。

卒業研究の御指導を頂いて以来、常に圧倒的な才能と知識をもった先輩として、絶えず私に懇切丁寧な御助言を下さり、啓発し、計り知れない影響を与えて下さった、京都大学工学部永持仁助教授に深く感謝いたします。

本研究は日本電信電話株式会社における研究業務の一環として行なわれたものです。入社以来7年間、直属の上司として、暖かく、時には厳しく御指導頂いた、現NTT長距離事業本部ネットワーク総合技術センタの井上明也博士に心からお礼を申し上げます。本研究の主な成果はNTT通信網研究所通信トラヒック部長の川島幸之助博士ならびにグループリーダの松田潤博士の部下である期間になされたものであり、両氏の御理解と御配慮が無かったならば、本研究の完成は大幅に遅れていたことでしょう。ここに深く感謝の意を表します。また、第4章の研究に対し、適切な御指導を頂き、絶大な影響を与えて下さった、NTT通信網研究所通信品質研究部長の間瀬憲一博士ならびにNTTサービス生産企画部担当部長の山本尚生博士に深謝致します。

私の愚問に対しても、常に懇切なる御返事を賜り、さらに研究方針など適切な御助言を頂いた、豊橋技術科学大学の増山繁助教授に厚く御礼申し上げます。また、神戸商科大学の加藤直樹教授、ならびに奈良先端科学技術大学院大学の福島雅夫教授にも、御忙しい中、度々御指導頂きました。ここに深く感謝いたします。

そして、本研究を取り纏める機会を与えて下さった、所長の青木利晴博士を始めとしたNTT通信網研究所の方々に感謝いたします。

最後に、常に私を支え、励まし、勇気付けてくれた、最愛の妻伸江に心から感謝致します。

目 次

| | |
|-----------------------|----|
| 謝辞 | i |
| 目次 | ii |
| 第1章 序論 | 1 |
| 1.1. 最適化問題とグラフ | 1 |
| 1.2. 通信網の基本構造とその管理 | 6 |
| 1.3. 通信網管理と最適化問題 | 9 |
| 1.4. 計算量理論 | 12 |
| 1.5. 本研究の位置付け | 16 |
| 第2章 領域を考慮した連結性問題 | 19 |
| 2.1. 通信網と連結性 | 19 |
| 2.2. 最小NA連結問題 | 23 |
| 2.3. 最小k-NA連結問題 | 32 |
| 2.4. T -混合カット上のNA連結 | 42 |
| 2.5. 最小直NA連結問題 | 45 |
| 2.6. まとめ | 61 |
| 第3章 制約付き多品種流問題 | 63 |
| 3.1. 通信網と多品種流問題 | 63 |
| 3.2. 経路枝数制限の付いた問題 | 65 |
| 3.3. 均等配分制約の付いた問題 | 76 |
| 3.4. まとめ | 95 |

| | | |
|------|-----------------------|-----|
| 第4章 | 迂回候補群作成問題 | 97 |
| 4.1. | 通信網制御の問題点とSTR | 97 |
| 4.2. | 迂回候補群作成問題と積上げ法 | 100 |
| 4.3. | 線形計画法に基づく迂回候補群作成法との比較 | 111 |
| 4.4. | 最小負荷経路選択法との比較 | 116 |
| 4.5. | まとめ | 118 |
| 第5章 | 結論 | 119 |
| 文献 | | 121 |
| 付録 | | 128 |
| 付録1 | 補題3.1の証明 | 128 |
| 付録2 | 定理3.4の証明 | 133 |
| 付録3 | 回線留保 | 135 |
| 付録4 | 繰り返し法のアルゴリズム | 136 |

第1章

序論

1. 1. 最適化問題とグラフ

グラフ、ネットワークに対する最適化問題は実用上重要な問題であり、様々な研究がなされ、多くの分野に応用されている。最適化問題は数学的に次の様に定式化される。

$$\begin{aligned} \text{目的関数(objective function): } f(x) \rightarrow \text{最小} \\ \text{制約条件(constraint): } x \in S \end{aligned} \tag{1.1}$$

ただし、 x は決定変数 (decision variable, 以下では単に変数と呼ぶ) の n 次元ベクトル $x=(x_1, x_2, \dots, x_n)$ であり、 S はこの問題(1.1)において変数 x がとることを許された値の集合であり、問題(1.1)の**実行可能領域(feasible region)**と呼ばれる。そして目的関数 f は S を含む適当な集合上で定義された実数値関数である。制約条件 $x \in S$ を満たす x を問題(1.1)の**実行可能解(feasible solution)**と言い、任意の $x \in S$ に対して $f(x^*) \leq f(x)$ を満たす実行可能解 $x^* \in S$ を問題(1.1)の**最適解(optimal solution)**と言う。最適化問題は具体的に与えられた f と S に対して、最適解 $x^* \in S$ を見いだすことが目的である。(最適解は一般には複数存在するが、通常はそのうち一つ見いだせば良いとされる。ただし、最適化問題の中には、最適解の全列挙を要請する様な問題もある。) また、目的関数が与えられず、実行可能解が存在する(yes)か否(no)かを判定する(存在するならば、その内の一つを見いだす)ことを要請される問題のことを特に**決定問題(decision problem)**という。

最適化問題は変数が整数等の離散値(かつ有限範囲に限定されることが多い)を取る場合には特に**組合せ最適化問題(combinatorial optimization problem)**と呼ばれる。組合せ最適化問題には、実に様々な実用上重要な問題が含まれるが、特にグラフ、ネットワークに対する問題はその中でも最も有名かつ重要な分野の一つであると言っても過言では無い。

以下、いくつか記号と用語の定義をおこなった後、グラフと関連する用語の定義を行なう。

記号 $:=$ は左辺を右辺で定義すること、あるいはアルゴリズム中では左辺の変数に右辺の数値を代入することを意味する。 $\lfloor x \rfloor$ は実数 x 以下の最大の整数、 $\lceil x \rceil$ は実数 x 以上の最小の整数を表す。集合 C の要素数を $|C|$ で表す。集合 C_1, C_2, \dots, C_n が集合 C の分割であるとは、

$$\bigcup_{i=1,2,\dots,n} C_i = C \quad \text{かつ} \quad C_i \cap C_j = \emptyset \quad (i \neq j)$$

を満たすことを言う。(但し $C_i = \emptyset$ であることも認める。)

グラフ(graph) $G=(V,E)$ は節点(node)の有限集合 V と、節点对を結ぶ枝(edge)の有限集合 $E \subseteq V \times V$ によって定義される。各枝はその両端の節点对を用いて $e(x,y)$ (但し $x,y \in V$) の様に表すことが出来る(但し、節点 x,y の間に枝が複数存在することが許される場合には、混同の怖れが無い場合に使用する)。この時、節点 x,y は枝 $e(x,y)$ の端点(terminal)であると言い、枝 $e(x,y)$ は節点 x,y に接続する(incident)と言い、 x と y は互いに隣接する(adjacent)という。枝に方向を考えない様なグラフを無向グラフ(undirected graph)、方向を考え $e(x,y)$ と $e(y,x)$ を区別する様なグラフを、有向グラフ(directed graph)という。有向グラフの枝を無向グラフの枝と区別するために特に有向枝(arc)と呼ぶこともある。有向枝を特に $a(x,y)$ と表現することもある。また、 $a(x,y)$ の両端点を区別して、 x を $a(x,y)$ の尾(tail)、 y を $a(x,y)$ の頭(head)と呼ぶこともある。また、有向枝の集合を記号 A を用いて表し、有向グラフを $G=(V,A)$ と表し、無向グラフと区別することもある。グラフが無向であるか有向であるかが文脈から明らかな時には単にグラフと記することもある。以下にグラフの基本的な用語を定義しておくが、これらは特に断わらない場合は無向グラフ、有向グラフ双方に適用される。なお、1要素からなる集合 $\{x\}$ は単に x とも書く。

グラフ $G=(V,E)$ に対して、グラフ $G'=(V',E')$ 、 $V' \subseteq V$ 、 $E' \subseteq E$ を G の部分グラフ(subgraph)という。 V 自身を節点集合とする(つまり $V'=V$ である)部分グラフを全域部分グラフ(spanning subgraph)という。 $E' \subseteq E$ に対して

$$V(E') := \{x,y \in V \mid e(x,y) \in E'\},$$

$V' \subseteq V$ に対して

$$E(V') := \{e(x,y) \in E \mid x,y \in V'\}$$

と定義する。 $G(V(E'),E')$ ($G(V',E(V'))$) を E' (V') で誘導されたグラフ(induced graph)と言う。 $W \subseteq V$ に対し、

$$G-W:=(V-W, E(V-W)),$$

$F \subseteq E$ に対し、

$$G-F:=(V, E-F)$$

と定義する。また、 $V(G)$ と $E(G)$ は、それぞれ G の節点集合と枝集合を意味する。2つの節点部分集合 $X, Y \subseteq V$ ($X \cap Y = \phi$) に対し、

$$E(X, Y) := \{e(x, y) \in E \mid x \in X, y \in Y\}$$

と定義する。グラフを明示したいときには $E(X, Y; G)$ とかく。全ての節点对 $x, y \in V$ に対して $|E(x, y)| \leq 1$ であるグラフを単純グラフ(simple graph)と言う。 $|E(x, y)| > 1$ であることも許すグラフを多重グラフ(multi graph)と言う。なお、上記の記号 $E(V')$, $E(G)$, $E(X, Y)$ 等は、有向グラフに話を限定している場合は、 $A(V')$, $A(G)$, $A(X, Y)$ 等と表す。

グラフ $G=(V, E)$ と節点部分集合 $W \subseteq V$ に対して、グラフ $G'=(V', E')$ が以下の条件を満たすとする。

- (1) $V' = V - W \cup \{w\}$ (但し w は V に含まれない新しい節点)
- (2) 任意の $x, y \in V - W$ に対し、 $|E(x, y; G)| = |E(x, y; G')|$ が成立する。
- (3) 任意の $x \in V - W$ に対し、 $|E(x, W; G)| = |E(x, w; G')|$ が成立する。(有向グラフの場合は $|A(W, x; G)| = |A(w, x; G')|$ も成立する。)

このとき G' を、 G の W による縮約グラフ(contracting graph)といい、 G/W で表す。また、縮約グラフを作る操作を縮約する(contract)という。

$$\deg(x) := |E(x, V-x)|$$

を節点 x の次数(degree)と言う (グラフを明示したい時には $\deg(x; G)$ と表す)。但し有向グラフの場合は、 x を尾とする枝と頭とする枝を区別して、

$$\deg_{\text{out}}(x) := |A(x, V-x)|$$

を節点 x の出次数(out-degree)、

$$\deg_{\text{in}}(x) := |A(V-x, x)|$$

を節点 x の入次数(in-degree)と呼ぶ。

節点と枝の交互列

$$v_1, e_1, v_2, e_2, \dots, v_{k-1}, e_{k-1}, v_k \quad (\text{但し、} e_i = e(v_i, v_{i+1}), i \in \{1, 2, \dots, k-1\})$$

を路(path)と言う。また、 v_1, v_k 間の路とも言う。 v_1 を始点、 v_k を終点と呼び、始終点を区別しないときには端点(terminal)と呼ぶこともある。 $X, Y \subseteq V$ について、 X, Y 間の路とは、 $x \in X, y \in Y$ 間の路のことである。始終点が一致する路を閉路(cycle)と言う。始終点以外は同じ節点を2度以上経由しない閉路は単純(simple)であると言う。路の長さ(length)は路に含まれる枝の数で

定義する。路を記述する場合は、含まれる節点または枝の系列として表現する。例えば

$$\langle v_1, v_2, \dots, v_n \rangle, \text{ 但し } v_1, v_2, \dots, v_n \in V$$

又は

$$\langle e_1, e_2, \dots, e_n \rangle, \text{ 但し } e_1, e_2, \dots, e_n \in E$$

の様に書く。また、路 p の節点集合を $V(p)$ 、枝集合を $E(p)$ （有向グラフに話を限定している場合は $A(p)$ ）と表現する。

グラフ $G=(V,E)$ において、 $x,y \in V$ 間に路が存在する時、 x と y は連結 (connected)であると言う。無向グラフ $G=(V,E)$ の部分グラフ $G'=(V',E')$ が、(1) $G'=(V',E')$ は連結であり、かつ(2) G' を真に含む（「 $V' \subseteq V''$ かつ $E' \subseteq E''$ 」かつ「 $V' \neq V''$ または $E' \neq E''$ 」）任意の部分グラフ $G''=(V'',E'')$ が連結ではないとき、 G' を G の連結成分 (connected component)という。無向グラフ $G=(V,E)$ において全ての節点对 $x,y \in V$ が連結である時、 G は連結であると言う。閉路を含まない無向グラフを森 (forest)といい、連結な森を木 (tree)という。無向グラフ $G=(V,E)$ の全域部分グラフ $G'=(V,E')$ で、 G' は森でありかつ、任意の $e \in E-E'$ について $(V,E' \cup e)$ が森ではない様なものを、 G の全域極大森 (maximal spanning forest)という。全域極大森が連結であるならば全域木 (spanning tree)という。明かに、連結なグラフの全域極大森は全域木である。

無向グラフ $G=(V,E)$, $x,y \in V$, $k \in \{\text{自然数}\}$ とする。任意の $W \subseteq V$, $|W| < k - |E(x,y)|$ に対して、 x,y が $G-W-E(x,y)$ において連結である時、 x と y は k -点連結 (k -node-connected)であると言う。 $G=(V,E)$ において任意の節点組 $x,y \in V$ が k -点連結ならば G は k -点連結であると言う。 k -点連結性を満たす最小の k を点連結度 (node-connectivity)と言ひ、 $x,y \in V$ 間の点連結度を $\kappa(x,y;G)$ （誤解の怖れの無い時には $\kappa(x,y)$ ）、 G の点連結度を $\kappa(G)$ と表す。 k -点連結、点連結度のことを単に k -連結 (k -connected)、連結度 (connectivity)と言うこともある。

無向グラフ $G=(V,E)$, $x,y \in V$, $k \in \{\text{自然数}\}$ とする。任意の $F \subseteq E$, $|F| < k$ に対して、 x,y が $G-F$ において連結である時、 x と y は k -枝連結 (k -edge-connected)であると言う。 $G=(V,E)$ において任意の節点組 $x,y \in V$ が k -枝連結ならば G は k -枝連結であると言う。枝連結度 $\lambda(x,y;G)$, $\lambda(x,y)$, $\lambda(G)$ も点連結度と同様に定義される。定義から、連結と1点連結と1枝連結は同値である。

複数本の路について、それらのどの二つの路も節点を共有しない時、これらの路は点素 (node-disjoint)であると言ひ、どの二つの路も枝を共有しない

い時、これらの路は枝素(edge-disjoint)であると言う。また、端点を除いて節点を共有しない時、これらの路は内素(internally-disjoint)であると言う。定義より、点素あるいは内素ならば、枝素である。

$G=(V,E)$ における $x,y \in V$ の距離(distance)とは、長さ最小の x,y 間の路の長さで、 $\text{dis}(x,y;G)$ あるいは $\text{dis}(x,y)$ と表す。 $\max_{x,y \in V} \text{dis}(x,y;G)$ を G の直径(diameter)と呼ぶ。

グラフはその定義の素朴さに比べ、実に様々な対象がモデル化されるため、とても重宝されている概念であるが、特に通信網や交通網等網状の形態をモデル化するのに有用である。次節で本論文で扱う通信網について説明する。

1. 2. 通信網の基本構造とその管理

通信網は現代社会ではもはや必要不可欠のものとなった。近年、急速な計算機技術の発展や、それを応用した高能力の交換機や設備の導入、情報伝達網の発展に伴い全国の情報監視が可能になる等によって、通信網において技術的に高度な制御、管理が行なえる様になってきた。

通信網の基本的な構造は、通信端末と交換機、加入者線、中継線からなる。加入者線と中継線を総称して通信回線（あるいは単に回線）という。通信端末（以後単に端末と記す）は加入者線によって交換機に接続され、適当な交換機間が中継線で接続される。通常、端末間を直接結ぶ通信回線は無い。よって、ある端末からある端末へ通信要求があると、適当な交換機を経由して接続される。各経由交換機では、適当な入回線と出回線とを接続する。これを交換と呼ぶ。交換方式には回線交換方式(circuit switching system)と蓄積交換方式(store and forward switching system)がある[AKIY 73]。回線交換方式は各通話が通話開始から終了まで回線を1本補足し続ける方式で、公衆電話回線のほとんどはこの方式である。蓄積交換方式交換方式は、通信を小包の様に断片的なデータに分け、バッファメモリに蓄積し、回線の空きを狙って送られる方式で、データ通信等の遅延があまり問題とならない場合の通信に適している。

公衆通信網は一般に大規模になるので、どの国でも公衆通信網は通常階層構造をしている。例えばNTTの公衆電話網では、各端末（電話機等）から延びている電話線は数千～数万本単位でGC(Group unit Center)と呼ばれる局に集約され、さらにGCから延びている通信回線はZC(Zone Center)に集約される。ZCは大体、県と一対一対応している。よって県間の通話がZC間の通話となると考えて良い。ZCと、ZC間の通信回線より成る通信網を中継網と呼ぶ（図1.1）。本研究で扱う問題は回線交換方式の公衆中継網上の問題が元になっている。本稿では今後単に中継網と記した場合は回線交換方式の公衆中継網を意味する。各ZCは通常複数の交換機によって構成されている。ZCの交換機を中継交換機、GCの交換機を加入者交換機と区別して呼ばれているが、本稿では、前述の様に基本的に中継網を対象とするので、単に交換機と言った場合、中継交換機のことを意味することにする。またZCのことを局(center)という。

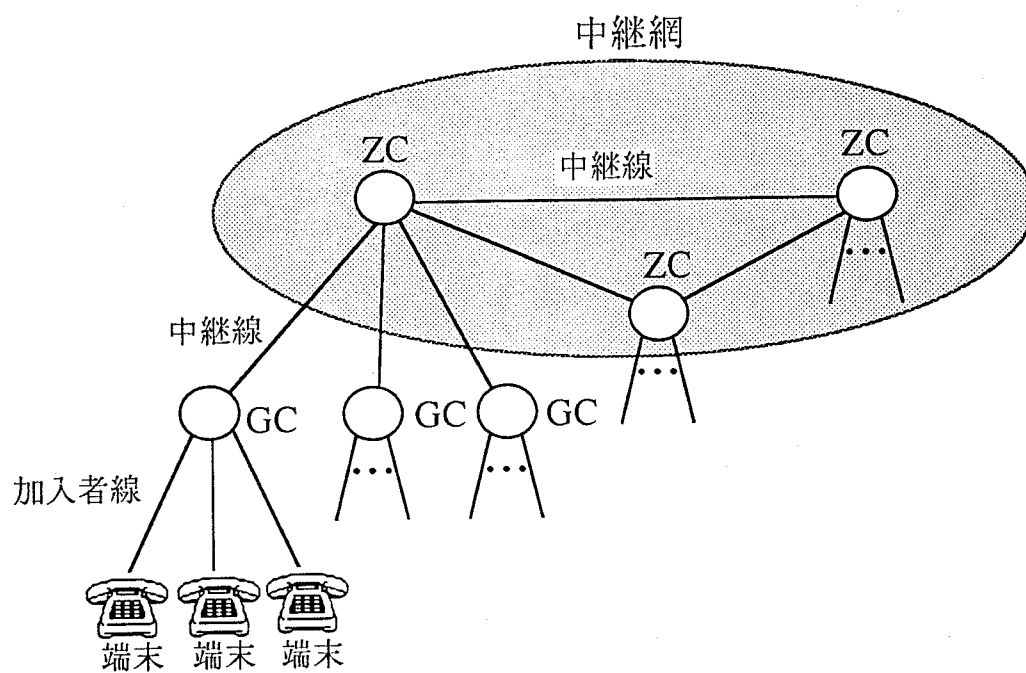


図1.1 NTT電話網の構造

Fig. 1.1 The Structure of NTT's telecommunication network.

通信網の管理(management)は設計(design)と制御(control)に大別される。設計は各局間の通信量（これをトラヒック(traffic)と呼ぶ）の需要を予測し、必要となる設備数を算出する作業である。設計結果を通信網に実現するには工事期間が必要なので、設計は数か月から一年の周期で実行される。制御は、現在の通信網設備を巧く運用することでなるべく良いサービスを提供し、サービス水準を維持しようとするものである。制御法には、通信要求をなるべく空いている回線を使って接続する迂回制御(routing control)、チケット予約等でよくあることだが通信網では処理仕切れない程の多量の通信要求があった時に、処理出来る量以上の通信要求を受付けなくする輻輳制御(congestion control)等がある。

こうした設計、制御を正確に行なうためには、通信網の状態やトラヒックを正確に把握するための、情報収集設備が必要である。NTTではATOMICSと呼ばれる収集装置によって、全国に300あまり存在する交換機から、受付呼（通信要求の一つ一つを呼(call)という）数や完了（接続に成功したことを完了(completion)という）呼数、呼の平均保留時間(average holding time; 回線を補足している時間)等のデータを3分毎に収集、蓄積し、全国の網管理局でそのデータを参照できる様にしている。その結果、通信網やトラヒックの現状の正確な把握と、それに基づいた素早い制御措置の実行、過去のデータの蓄積に基づいたより緻密な設計が出来る基盤が整ってきている。例えば、伝送路が故障で切れた場合には、これまでは管理者が限られたデータと長年の感に基づいて切り替え先の予備伝送路を決定していたが、これからは全国の伝送路網データと現在の使用状況、これからの工事予定等を考慮に置いて、最適な切り替え先を決定できるようになる。

近年になって通信網管理に影響を及ぼしたもう一つの要因に、使用者の意識の変化がある。つまり、新サービスの導入や複数会社の競合、電話の使用目的の多用化、料金体形の変化等によってトラヒック変動も多様化したため、逆に日々のトラヒック予測が困難になりつつある。しかし通信網を管理するには、このような状況においても網設備の過不足を無くし、常に効率的に網を運用していく必要がある。そのためには、前述のATOMICSのような情報収集装置を用いて得られた正確なデータに基づき、計算機を用いて高速に最適化問題を解く必要が増えてくる。ではどのような最適化問題が必要とされるのか、次節で説明する。

1. 3. 通信網管理と最適化問題

前述の様に通信網はグラフによってモデル化し易い。例えば、中継回線を枝に、交換機を節点に置き換えることによって、電話網の接続関係がグラフによって簡単に表現される。勿論、こうしてモデル化することによって、各回線の伝送方式の違い、各交換機の規模や性能の違い等、多くの情報が失われるが、逆にそういった情報とは無関係な性質（例えば任意の交換機間で通信が可能か否か等）を扱う場合には、却ってこうした簡単なモデルの方が扱い良い。

通信網の設計、制御を考える場合には単に接続してるだけではなく、各節点間でどれだけの情報をやりとりできるかを考慮することも重要である。そのため、グラフの枝に容量を与え、指定した節点間に指定した量の流れを流せるか否かを扱う様なモデルを考へることも意味深い。この様なモデルを通常ネットワーク(network)と呼ぶ。ただし、ネットワークの定義は扱う問題によって異なり、統一した定義は無い。本研究では第3章と第4章でネットワークを扱い、定義もその都度行なう。

グラフ、ネットワークに対する組合せ最適化問題には様々な問題があり、通信網設計・制御に貢献している。以下にその例を上げる。

最小費用木問題(MST; minimum cost spanning-tree problem)

[入力] 連結無向グラフ $G=(V,E)$ 、枝の費用関数 $c: E \rightarrow \{\text{正実数}\}$

[目的関数] $\sum_{e \in E} c(e) \rightarrow \text{最小}$

[制約条件] $G'=(V,E')$ は G の全域木 (木である全域部分グラフ)

最大流問題(MF; maximum flow problem)

[入力] 有向グラフ $G=(V,A)$ 、流出点 $s \in V$ 、流入点 $t \in V$ 、

枝の容量関数 $u: A \rightarrow \{\text{正実数}\}$

[目的関数] $\sum_{a \in A(s, V-s)} f(a) - \sum_{a \in A(V-s, s)} f(a) \rightarrow \text{最大}$

[制約条件] $0 \leq f(a) \leq u(a), a \in A$

$$\sum_{a \in A(v, V-v)} f(a) - \sum_{a \in A(V-v, v)} f(a) \begin{cases} =0, v \in V - \{s, t\} \\ \geq 0, v=s \\ \leq 0, v=t \end{cases}$$

$f: A \rightarrow \{\text{正実数}\}$

線形計画問題(LP; linear programming problem)

[入力] $a_{ij}, b_i, c_j \in \{\text{実数}\}, i=1,2,\dots,m, j=1,2,\dots,n$ (但し n, m は自然数)

[目的関数] $\sum_{j=1}^n c_j x_j \rightarrow \text{最小}$

[制約条件] $\sum_{j=1}^n a_{ij} x_j = b_i, i=1,2,\dots,m$

$x_j \geq 0, j=1,2,\dots,n$

最小費用木問題MSTにおいて、変数を、 $|E|$ 次元のベクトル（各成分が各枝に対応する）と考えて、解に含まれる枝に対応する変数を1に、解に含まれない枝に対応する変数を0とすると考えると、MSTが組合せ最適化問題であることが容易に理解される。最大流問題MFは、変数 f の定義域が正実数であるが、容量関数が整数であるならば最適解も整数ベクトルになることを保証する定理[FORD 59] [IWAN 93]があるので、組合せ最適化問題と考えることが出来る。また、容量関数が実数の場合でも、線形計画問題LPに定式化できるため、次に述べる様にLPが組合せ最適化問題と考えられるので、やはり組合せ最適化問題となる。LPも変数 x_{ij} の定義域が実数であるが、最適解が存在するならば、実行可能領域を表す n 次元凸多面体の端点に最適解が存在することが保証されており[DANT 63] [IBAR 93]、やはり組合せ最適化問題と考えることが可能である。

これらの、通信網に対する適用例を説明する。通信網の骨格を構成する問題がMSTに定式化される。骨格とは最低限存在しなければならない回線を意味する。骨格にさらに経済性を考慮して回線を付与していくことで、通信網の形ができる。通信網は、交換機を節点、通信回線を枝に対応さ

ることでグラフに定式化できる。通信網の満たすべき最低限の条件として、任意の交換機間に通信経路が存在することであろう。これを満たす必要十分条件がグラフが連結であることである。通信回線を張るのに要する費用をその枝の費用とすれば、最小費用な連結全域部分グラフを作成する問題になる。費用関数が正なので、最小費用連結全域部分グラフは最小費用全域木になり（詳細な説明は略する）、本問題がMSTになることが分かる。現在の空き設備で、指定した2局間にどれだけの通信が可能かを計る場合にはMFが使われる。局を節点、局間の通信回線を枝に対応させ、使用可能な回線数を枝の容量とすれば良い。LPはその定義の広さから様々な問題が定式化される。代表的な例としてAT&Tが通信網設計を、大規模LPを解く算法として有力視されているKarmarkar法[KARM 84]を用いて行なっていた時期があった[ASH 81]ことをあげておく。

1. 4. 計算量理論

本論文では、計算量理論が重要な役割を果たすので、本節でその概略を説明する。詳細は[GARE 79] [PAPA 82] [IBAR89b]等の専門書を参照されたい。組合せ最適化問題には1. 3で述べたMST, MF, LPの他にも様々な問題(problem)がある。与えられた問題を解く為の、機械的操作からなる有限の手続きをアルゴリズム(algorithm)という。但しアルゴリズムはその実行が有限時間内に停止しなければならない。必ずしも有限時間に停止することが保証されていないものを、単に手続き(procedure)と呼んで、アルゴリズムと区別する。アルゴリズムの評価法で重要なのは、その実行に必要な計算時間と記憶領域の広さである。計算時間や記憶領域は、実際には使用する計算機やプログラミング言語の違いで異なるが、組合せ最適化理論では、これを計算量(complexity)という概念に統一して扱う。計算量には時間量(time complexity)と領域量(space complexity)とがあり、それぞれが計算時間と記憶領域に対応している。本論文では、時間量のみを扱うが、その理由は、通常、領域量は時間量で押さえられるからである。よって今後、特に断わらない限り、計算量とは時間量を意味するものとする。計算量とは、計算機にランダムアクセスマシン(RAM)を想定し、四則演算や比較等の基本演算を1ステップと考えたときの、計算終了までに必要なステップ数である。一つの問題は、通常無限個の問題例(problem instance)からなっている。例えば問題MSTは、グラフ $G=(V,E)$ と枝のコスト関数 $c: E \rightarrow \{\text{正実数}\}$ を具体的なデータとして与えることによって定まる無数の問題例の集合である。同じアルゴリズムでも、問題例の規模が大きくなれば、計算に要するステップ数が大きくなるので、ステップ数は問題例との関連下に評価しなければならない。そこで、アルゴリズムの計算量は、問題例の入力(つまり問題例の特定)に必要なビット数 n の関数 $T(n)$ で表現する。つまり、ある問題のアルゴリズムが、入力長 n 以下の問題例全てを $T(n)$ ステップ以下で解く時、そのアルゴリズムの計算量は $T(n)$ であると言う。計算量理論において通常は定数倍程度の差は本質的では無いため、お互いがお互いの定数倍以下に収まる様な関数を一纏めにして $O(f(n))$ (オーダー $f(n)$ と読む)の様に表現する。つまり、 $T(n)=O(f(n))$ であるとは、ある定数 c, m が存在して、全ての $n \geq m$ に対して

$$T(n) \leq cf(n)$$

が成立することを意味する。 $O(f(n))$ の $f(n)$ にはなるべく簡単な関数を使う。
例えば

$$2.5n^2 + n \log n + 100n = O(n^2)$$

となる。

アルゴリズムの実用性を判断する際に、計算量が多項式であるか指数関数になっているかが本質的な違いとされる。組合せ最適化問題の多くは、入力長 n に対し、解の候補が高々 $O(a^n)$ (a はある定数) 個であり、それらを列挙して一つ一つ調べていけば有限時間内に解ける。この様な問題の集合をNPと呼ぶ(但しNPの厳密な定義は[GARE 79]参照)。クラスNPに属する問題はこのような列挙法を用いれば指数オーダーの計算量で解けるが、 n の増加に伴って計算量が爆発的に増加し、有限ではあっても実際問題として解くのは不可能である。そこで、多項式オーダーの計算量で解くアルゴリズム(これを多項式時間アルゴリズム(polynomial-time algorithm)という)が存在する問題の集合をPと記し、各問題がクラスPに属するかどうかを、その問題が実際問題として解けるのかどうかの判定基準と考えるのである。勿論、多項式時間アルゴリズムでも、その次数が大きければ(例えば $O(n^{10})$ 等)実際に解くのは難しいが、たとえ次数が大きくとも一旦多項式オーダーのアルゴリズムが発見されてしまうと、それが改良されて実用に耐える低い次数のアルゴリズムが開発されていくことが多い。

以下本節では決定問題(1. 1参照)に話を限定するが、一般の最適化問題に拡張しても同様の議論が成立する。クラスNPに属する全ての問題がクラスPに属するかどうか(つまり $P=NP$ かどうか)は未解決問題である。しかしNPの中で最も難しい問題、つまりこの問題がクラスPに属するならば $P=NP$ が成立するという問題の集合が見つかっている。これをNP完全(NP-complete)問題と呼ぶ。また、どのNP問題よりも同等以上の難しさを持つ問題をNP困難(NP-hard)であると言う。NP困難かつクラスNPに属するならば、NP完全である。第一号のNP完全問題として、充足可能性問題(SAT)がNP完全であることが[COOK 71]によって証明された。SATがNP完全問題であることが判明すると、それ以降のNP完全性の証明には帰着可能性(reducibility)という概念を用いることができる。すなわち

問題Aの任意の問題例 X に対し、多項式時間で計算できる関数 g によって、問題Bの問題例 $g(X)$ を構成でき、Aにおける X の答(yesあるいはno)とBにおける $g(X)$ の答えが一致するとき、AはBに多項式的に

(polynomially)帰着可能である。

以下本稿では、多項式的に帰着可能なことを単に帰着可能という。既知のNP完全問題Aが、クラスNPに属する問題Bに帰着可能であることが証明できたならば、問題BがNP完全であることが証明できる。この手法を用いることによってNP完全問題が次々と発見されていき、これまでに数えきれない程見つかった[GAARE 79]。これら全てに対して今だに多項式時間アルゴリズムが見つからないので、今では大部分の研究者が $P \neq NP$ と予想している。

NP完全問題の中には、一見多項式時間に見える計算量をもつアルゴリズムが見つまっているものもある。例えば「 n 個の自然数を入力とし、それらを2つの集まりに分割して、それぞれの大きさ（含む要素の大きさの和）を全く同じにできるか？」という問題は分割問題(PARTITION)と呼ばれるNP完全問題であるが、これは各要素の大きさの和を K とすると、 $O(nK)$ で解くことができることが分かっている[GAARE 79]。これがクラスPを定義する上での多項式時間アルゴリズムでは無い理由は、 K を入力するのに必要なビット数は $O(\log K)$ なので、 nK は、PARTITIONの問題に必要な入力長 $n + \log K$ の指数関数になっているからである。しかし、このアルゴリズムは K が小さい値であるならば、十分速いアルゴリズムとも言える。そこで、この様に問題例の入力に現われる数値に対する多項式まで認めたものを考える。すなわち、問題Aの問題例 X に対し、 X の入力に必要なビット数を $\text{length}(X)$ 、 X 中に現われる最大の数値（例えばPARTITIONにおける K ）を $\text{number}(X)$ と表すとする。 $\text{length}(X)$ と $\text{number}(X)$ の多項式で表される計算量を持つアルゴリズムを擬多項式時間(pseudo-polynomial time)アルゴリズムと言う。また、ある多項式 $f(n)$ が存在し、 $\text{number}(X) \leq f(\text{length}(X))$ の制約下においてもなお問題AがNP完全（困難）であるとき、問題Aは強(strongly)NP完全（困難）であると言う。 $P \neq NP$ の仮定の下では、強NP困難な問題には擬多項式アルゴリズムは存在しない。

さて、ある通信網管理上の問題を解決しようとした場合、まずそれを最適化問題に定式化する。多くの場合クラスNPに属する問題になる。よって次にすべきことはクラスPに属するかNP完全であるかを判定することである。（但し $P \neq NP$ の仮定の下では、PでもNP完全でも無いNP問題が無数にあることが証明されていて[LADN 75]、実際未だにPかNP完全か判明していないNP問題も存在するが、大多数の問題はどちらかであることが証明で

きる。)そしてクラスPに属することが分かれば、より効率的なアルゴリズムの開発を目指す。また、NP完全であるならば、高速な近似アルゴリズムの開発や、分枝限定法[IBAR 83]等を用いて最悪時の保証は無いが平均的に速いアルゴリズムの開発を試みる等で問題の解決を計る。近似(approximation)アルゴリズムとは、NP困難等最適解を得るのが困難である問題に対し、目標関数値が最適解に近い値の解を得ることを目的としたアルゴリズムである。目的関数を $f(x) \rightarrow$ 最小、最適解を x^* 、近似解を x' としたとき、

$$f(x') - f(x^*)$$

を絶対近似誤差(absolute error)、

$$f(x') / f(x^*)$$

を相対近似誤差(relative error)という。(但し、通常最適化問題では $f(x^*) > 0$ となる様に定式化される。相対近似誤差はこの不等式が成立する場合に意味を持つ。)近似アルゴリズムを開発するときには、絶対あるいは相対近似誤差が小さい定数で押さえられる様なアルゴリズムを得ることが一つの目標になる。また、

$$b \leq f(x^*)$$

を満たす実数値 b を下界値(lower bound)という。自明でない下界値を発見することで、近似誤差の評価ができることが多い。

1. 5. 本研究の位置付け

グラフ、ネットワーク上の組合せ最適化問題には、前節までに述べた他にも多くの重要な問題があり、それぞれについて詳しく研究がなされ、様々な効率的な解法が提案されている。しかし組合せ最適化問題は制約条件が少し変化しただけで、最適性の条件が変わり、元の解法が使えないことが多い。現実の通信網等への応用を考えると実に様々な制約条件や目的関数があり、前述の様な単純な問題を解くだけでは不十分であることが多い。

例えば、通信網の各回線群に必要な回線数は、そこを流れる呼量の非線形関数になる。例えば、呼の生起がランダム、保留時間の分布が指数関数であるという仮定の下では、回線数 n の回線束に、平均的に trf アーランの呼の要求があった時の呼損率（接続要求のあった呼のうち接続出来ない呼の割合） b は

$$b = B(n, trf) := \frac{trf^n/n!}{1 + \sum_{i=1}^n trf^i/i!}$$

となる。この式をアーランB式と呼ぶ[AKIY 73]。また、回線は網の種類によって異なるが24～600本を設定の最小単位としているので、回線束のコストは階段関数（単調増加かつ微分可能な点における微分係数が0）になる。このコスト関数を厳密に扱って最適化問題を解くのは大変難しい。

また、通信網の構造も厳密に考えると単純なグラフによるモデル化では不十分なことが多い。例えば現在の中継網はマルチユニット構成をとっている。それは、各ZCが複数個の中継交換機から構成されており、その内の任意の中継交換機から、そのZC内の任意の加入者交換機に接続が可能になっている。但し、同じZC内の交換機どうしでも、互いに接続しているものと接続していないものがある。よってこれをグラフでモデル化する場合、中継交換機を節点に対応させると、ZCの組分けの情報が失われ、ZCを節点に対応させると、連結性の情報が失われる。どちらか片方の情報があれば十分な場合は2つのモデル化法を使い分ければ良いが、目的によってはどちらのモデル化でも不十分な場合もある（この問題については2章で詳しく扱う）。

以上に挙げた例は一例であり、他にトラヒックの種類（電話、FAX、データ等）やサービスの違い（テレジョーズ、NCC(New Common Carrier)呼等）によるトラヒックの性質の違い、設備故障時の接続品質の保証等を全て考慮して最適化することはほぼ不可能である。

しかし現実には通信網を管理していく立場としては、不可能であると手をこまねいているわけにもいかない。そこでよく用いられる解法は、適当な初期値から初めて、その地点での適当な改善方向を決めて局所的に解を改良していき、ある条件を満足すれば停止するという、発見的な算法である。この方法は最適解に辿り着く保証は勿論無いが、計算が比較的容易であり、しかもある程度妥当な解（最適解に近い目的関数値を持つ解）に落ち着く可能性が高い。また、もう少し工夫をする場合には、実行可能解を効率的に列挙する方法である、分枝限定法[IBAR 83]を用いることもある。これも大規模な問題では解を全て列挙することは難しいので、適当な時間で停止させ、その時点で得られている最良の解を用いることになる。しかし、こういった方法をとる場合でも、なるべくその問題の本質を良く反映した最適化手法を取り入れることにより、解の近似の度合いや、計算時間が改善される。そこで、通信網管理に現われる制約条件で、これまで理論的に扱われておらず、しかも一般性があると思われる条件を抽出し、そこに的を絞って研究することで、従来の理論が応用出来、その上に新たな理論的知見が得られ、より現実のモデルに忠実な効率的解法の発見があることが期待される。これらの条件の内いくつかは、今までの概念の拡張になっている場合もあり、理論的发展に対する寄与にも期待が持てる。

本研究では、通信網管理から発生したグラフ、ネットワーク上の組合せ最適化問題について、一般性が有るものについて抽出し、検討した。その結果新たなモデル化法、効率的に解ける問題のクラス、解けない問題のクラス、効率的算法等を明らかにした。第2章では、通信網の骨格を設計する問題にマルチユニット等を考慮した場合のモデル化法を提案した。そして連結度を拡張した概念「節点・領域連結度」を提案し、その概念を扱う上で重要な性質をいくつか明らかにした。第3章では多品種のネットワークフロー問題に、通信網制御上よく現われる制約「経路枝数制約」と「均等配分制約」を加えた問題を考え、それらの計算量を明らかにした。そして、第4章では、それらの知識に基づいて、実際の通信網管理の問題「迂回候補群作成問題」を解いて、実用に貢献した。

本研究で扱った問題は、通信網管理において重要であることは勿論だが、通信網以外でも、交通流等多くの問題がモデル化されと考えられる。よってこの結果が、今後、最適化問題の実用研究に広く使われていくことを願う。

第2章

領域を考慮した連結性問題

2. 1. 通信網と連結性

通信網に関する諸現象を理論的に扱う場合、交換局を節点に、交換局間の回線群を枝に対応させてグラフにモデル化することが多い。グラフの強固さを測る指標として、連結度がある。連結度の意味を通信網で考えると、通信経路を確保するのに、いくつの交換局の故障まで耐えられるか、ということを表す。連結度はグラフ理論に古くから登場している概念であり、その性質について多くのことが明らかになっている[HARA 69] [BOLL 78]。また連結度を用いて通信網の耐故障性を評価して、より頑丈な網構成を提案するという研究もなされている[SONE 88]。

グラフに枝を付加すれば、付加した枝の両端点間の連結度は増加するし、グラフ全体の連結度も増加するかもしれないが、少なくとも減少することは無い。よって単に連結度を高めることだけが目的ならば、枝を増やせば増やすほど良いことになる。しかし、経済性や保守の容易性という観点から、なるべく少ない枝数で高い連結性を保持することが望ましい。よってこれまでも少ない枝数で高い連結度を達成しようとする研究が多い。

[DEBR 46] [KAUT 68] [SONE 88]らは限られた枝数で連結度が高くかつ直径の小さいグラフの構成法を提案した。これらはグラフ的な視点から理想的な網構成を提示したもので重要である。[FRAN 70] [ESWA 76] [FRAN 90] [NAOR 90] [WATA 93]らは与えられたグラフに最小数枝を付加することで所望の点連結度（あるいは枝連結度）を達成する多項式時間アルゴリズムを提案している。これらは任意の節点間に枝を付加して良いという原則の問題であるが、実際の通信網では地理的、経済的等の理由で回線を付加出来ない対地が多く存在することも事実である。したがって、こういった場合に対応するためには、枝を付加できる場所の制限がある様な問題も重要である。この条件下の問題についてもいくつか研究されているが、[FRED

81] では木に最小数の枝を付加して2点連結（あるいは2枝連結）にする問題でさえNP困難であることが示されている。そして[FRED 81] [WATA 92] では、グラフに何本かの枝を追加して、 k -点連結（あるいは k -枝連結）にする問題の近似アルゴリズムが提案されている。また、[HAN 92] [KHUI 92] [NAGA 92] [FRAN 93] [GARG 93]らは与えられた k -点連結（あるいは k -枝連結）なグラフから、 k -点連結（あるいは k -枝連結）を保ったまま枝を最大数削除する問題（NP困難）を扱い、それらの高速近似アルゴリズムを提案している。

グラフの k -連結性を確保する問題については、この様に種々の制約を考慮した検討がなされているが、実際の通信網の最適化に適用するにはまだ不十分な点がある。その一つとして、通信網では複数の交換機が集まって一つの論理的な領域（交換機の集まり）を形成していることがある。通信は、一つの論理的領域を目的地として経路を選択し、領域内のどれかの交換機に着信すれば、目的となる着端末まで通信路を確保することができる（図2.1参照）。この場合、着領域のどれか一つの交換機に接続しさえすれば良い。そこで交換機を節点に対応させてグラフにモデル化すれば（図2.1(b)参照）、各論理的領域は一つの節点集合で表される。その結果、重要なのは、節点間の連結度ではなく、節点と領域の連結度であると言える。

節点と領域の連結度は、[HARA 69]にあるように、領域を節点に縮約したグラフにおける節点間の連結度になるので、従来、自明なものとしてあまり理論的に扱われることは無かった。しかし領域を節点に縮約すると、他の節点間や節点と領域間の連結度が変わってしまうため（図2.2参照）、グラフ全体の連結度を議論する場合には、この方法は使えない。

以上の観点から、本章では、節点と領域の連結度に関し、種々の性質を明らかにする。まず2.2で節点と領域の連結性を意味する、NA連結 (node-to-area connection) という概念を導入する。そのあと、与えられたグラフと領域に対し、NA連結である最小枝数全域部分グラフを求める問題がNP困難であること、さらにこの問題に対する線形時間近似アルゴリズムを示す。この近似アルゴリズムの絶対近似誤差を見積り、本問題のNP困難な部分問題に対し、絶対近似誤差が定数であること[ITO 94b]を示す。2.3では k -NA連結性を定義し、その後NA連結に対する議論を k -NA連結に拡張し、いくつかの性質を明らかにする。そして与えられたグラフに対し、1-連結性を保存する全域部分グラフは、任意の節点・領域間に対し k -NA連

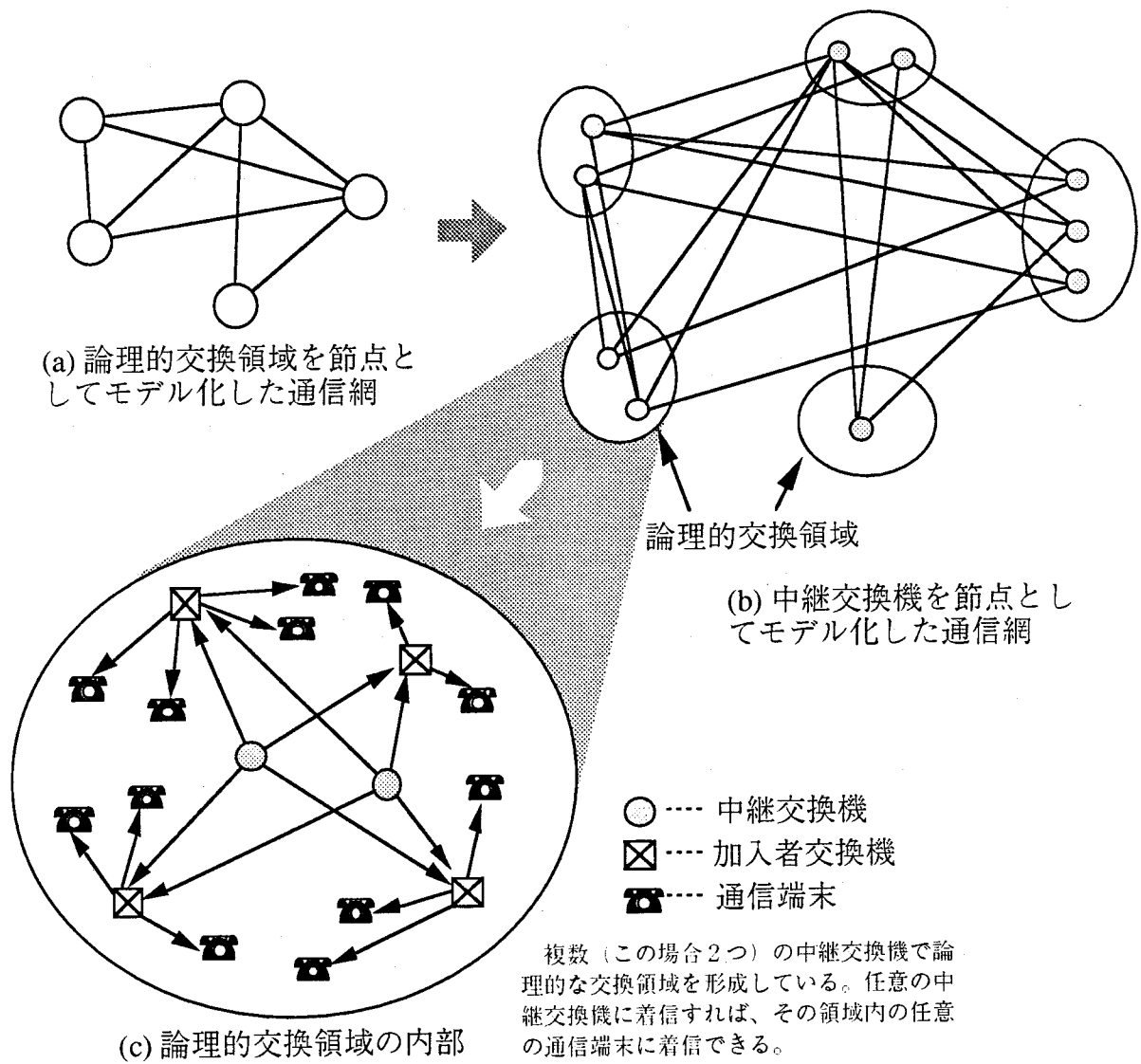
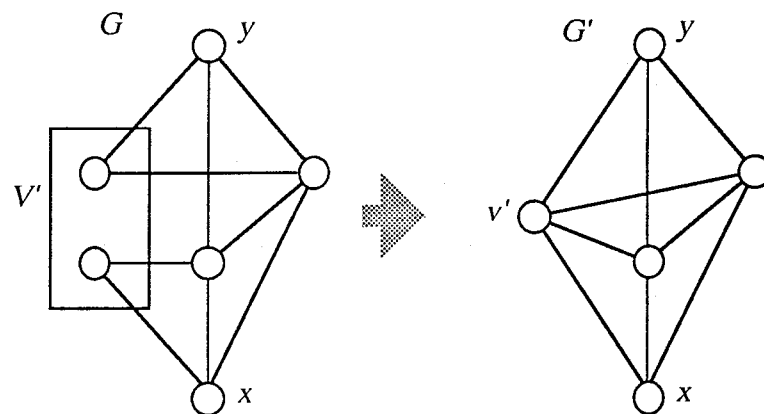


図2.1. 通信網の構成

Fig. 2.1. Structure of a telecommunication network and its modeling.

結性も保存すること [ITO 94b] を証明する。この性質を用いれば、 k -NA連結性を保存する最小全域部分グラフを求める問題 (NP困難) に対し、 k -連結に関して得られた高速アルゴリズム [NAGA 92] [FRAN 93] を適用することによって、相対近似誤差が2倍以内の解を線形時間で得ることができる。2.4では点連結と枝連結の一般化概念である、 T -独立な経路数の概念に基づく連結度にNA連結を適用し、点間の連結度とNA連結度との関係 [ITO 94c] を示す。2.5ではさらに大規模故障時の被災地からの通信確保を意識した制約条件を加えた問題について考察し、NP完全性を示すと共に、近似アルゴリズムを与える [ITO 94a] [ITO 95]。

なお本章では全て無向グラフを扱う。よって単にグラフ、枝と記してあるものは全て無向グラフ、無向枝を意味する。



グラフ G 上では x, y の連結度は2だが、 V を縮約して得られたグラフ G' 上では x, y の連結度は3。

図2.2. 縮約によって連結度が変わる例

Fig. 2.2. Contraction destroys connectivity.

2. 2. 最小NA連結問題

2. 2. 1. NA連結性

【定義】 グラフ $G=(V,E)$ に V の分割 $X=\{V_1, V_2, \dots, V_H\}$ が付与されたものを、領域グラフ(area graph)と呼び、 $G^a=(G,X)$ で表す。□

【定義】 グラフ $G=(V,E)$ において、 $V' \subset V$ と $x \in V - V'$ について、ある $y \in V'$ が存在して x と y が連結である時、 x と V' は節点・領域連結 (node-to-area connected; NA連結) であると言う。また、領域グラフ $G^a=(G,X)$ において、任意の V_i と任意の $x \in V - V_i$ について、 x と V_i がNA連結である時、 G^a は節点・領域連結 (NA連結) であると言う。□

【定義】 NA連結な領域グラフ $G^a=(G,X)$ に対し、 $G=(V,E)$ の部分グラフ $G'=(V',E')$ と $X'=\{V'_i=V_i \cap V' \mid V_i \in X\}$ について、領域グラフ (G',X') がNA連結であるとき、 G' は G^a (あるいは G) のNA連結な部分グラフであるという。□

定義より、グラフ G が連結ならば任意の領域 X に対し (G,X) はNA連結である。しかし、グラフ G が連結でないにも関わらず、領域グラフ (G,X) がNA連結である様な例が存在する (図2.3参照)。

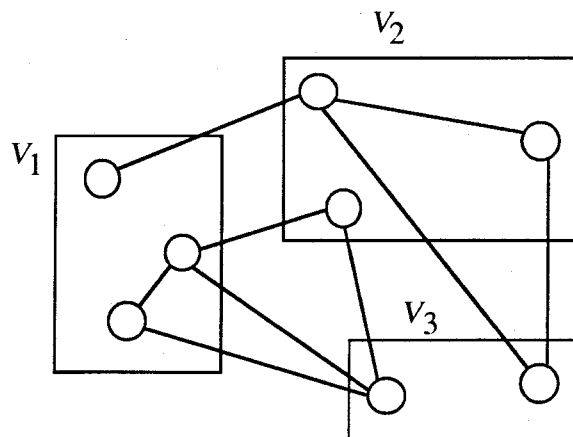


図2.3. 非連結だが、NA連結なグラフ

Fig. 2.3. Disconnected but NA-connected graph.

2. 2. 2. 最小NA連結問題の複雑さ

本節では次の問題を扱う。

【定義】 最小NA連結問題(Minimum NA-Connected subgraph problem: MNAC)

[入力] NA連結な領域グラフ $G^a=(G,X)$, $h \in \{\text{自然数}\}$

[要請] $|E'| \leq h$ である様な G^a のNA連結全域部分グラフ $G'=(V,E')$ が存在するかどうかを判定せよ。□

最小NA連結問題に対して次の定理が成立することを以下で示す。

【定理2.1】 最小NA連結問題はNP完全である。□

本定理の証明の前にNA連結性の必要十分条件を補題として示す。この性質は後述の定理2.2の証明にも使用する。

【補題2.1】 領域グラフ $G^a=(G,X)$ がNA連結である必要十分条件は、 G の各連結成分が、すべての領域 $V_i \in X$ 内の少なくとも一点を含むことである。

□

証明)

必要性 (「 G^a がNA連結」 \rightarrow 「各連結成分がすべての領域 $V_i \in X$ 内の少なくとも一点を含む」) の証明。

対偶を証明する。ある連結成分 G_j が、ある領域 V_i 内の節点を一つも含まないと仮定する。このとき G_j に含まれる節点 x を一つとると、どの $y \in V_i$ についても x と y は連結でない。つまり、 G^a はNA連結ではない。(必要性の証明了)

十分性 (「各連結成分がすべての領域 $V_i \in X$ 内の少なくとも一点を含む」 \rightarrow 「 G^a がNA連結」) の証明。

対偶を証明する。 G がNA連結でないと仮定する。すなわち定義から、ある $V_i \in X$ とある $x \in V - V_i$ に対し、どの $y \in V_i$ をとっても x と y が連結でない。これは x を含む連結成分を G_j とすると、 G_j は V_i に属する節点を一つも含まな

いことを意味する。(十分性の証明了) Q.E.D

定理2.1の証明) 最小NA連結問題MNACがNPに属するのは明らか。以下でNP困難性を示す。

既知のNP完全問題である充足可能性問題SAT[COOK 71] [GARE 79]をMNACに帰着する。

【定義】 充足可能性問題(SAT)

[入力] $U=\{u_1, u_2, \dots, u_N\}$: ブール変数集合

$C=\{C_1, C_2, \dots, C_M\}$: U 上の節の集合

[要請] 論理式 $C_1 \wedge C_2 \wedge \dots \wedge C_M=1$ を満足することが可能であるか否かを判定せよ。□

SATの問題例 (U, C) ($U=\{u_1, u_2, \dots, u_N\}$, $C=\{C_1, C_2, \dots, C_M\}$)に基づいて、MNACの問題例 (G^{na}, h) を以下のように構成し、両問題の解 (yesかnoか) が一致することを示す。

$$G^{\text{na}}=(G=(V, E), X)$$

$$V=\{u_1, u_2, \dots, u_N, \neg u_1, \neg u_2, \dots, \neg u_N, C_1, C_2, \dots, C_M, D_1, D_2, \dots, D_M, r_1, r_0\}$$

$$E=E_1 \cup E_2 \cup E_3$$

$$E_1=\{e(u_i, r_1), e(\neg u_i, r_1), e(u_i, r_0), e(\neg u_i, r_0) \mid i=1, 2, \dots, N\}$$

$$E_2=\{e(D_j, r_0) \mid j=1, 2, \dots, M\}$$

$$E_3=\{e(u, C_j) \mid u \in U \text{かつ} u \in C_j\} \cup \{e(\neg u, C_j) \mid u \in U \text{かつ} \neg u \in C_j\}$$

$$V_i=\{u_i, \neg u_i\}, i=1, 2, \dots, N$$

$$V_{N+j}=\{C_j, D_j\}, j=1, 2, \dots, M$$

$$V_{N+M+1}=\{r_1, r_0\}$$

$$h=2(N+M)$$

例としてSATの問題例

$$U=\{u_1, u_2, u_3, u_4\}, C=\{\{u_1, \neg u_3, \neg u_4\}, \{\neg u_1, u_2, \neg u_4\}\}$$

に対して構成したMNACの問題例を図2.4に示す。

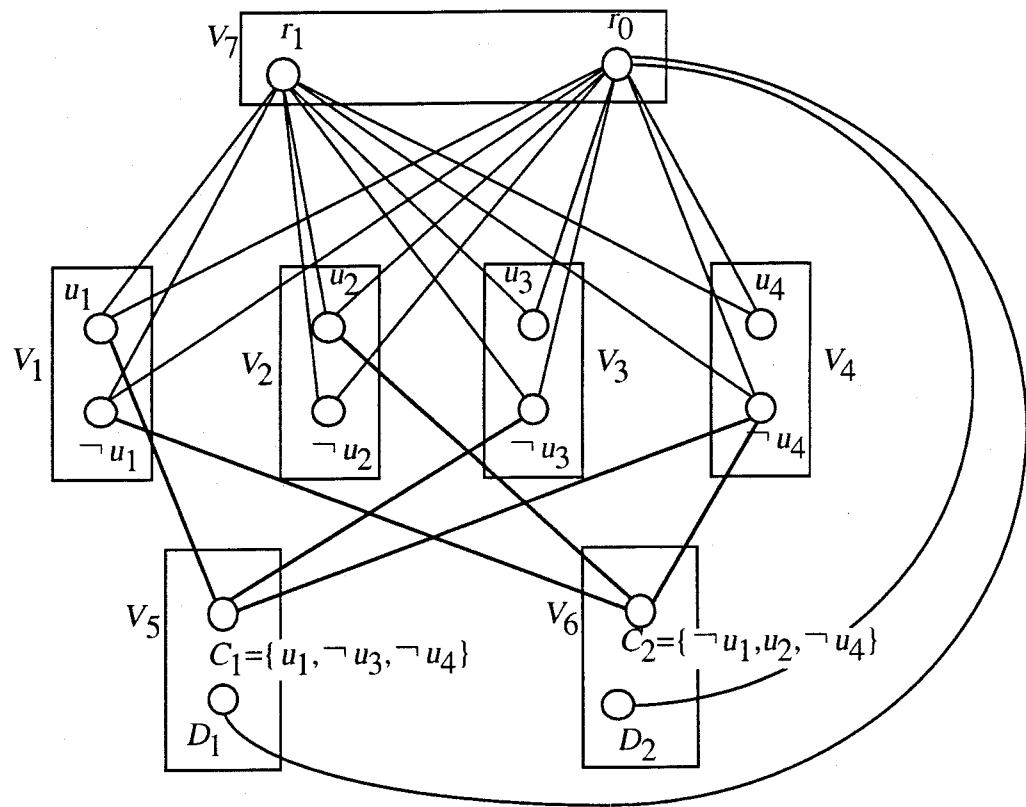


図2.4. SATからMNACへの帰着例
 SATの問題例は $U=\{u_1, u_2, u_3, u_4\}$, $C=\{\{u_1, \neg u_3, \neg u_4\}, \{\neg u_1, u_2, \neg u_4\}\}$

Fig. 2.4. Reduction from SAT to MNAC.

「MNACにおいて (G^a, h) がyes \rightarrow SATにおいて (U, C) がyes」の証明

G^a のNA連結全域部分グラフ $G'=(V, E')$ で $|E'| \leq h=2(N+M)$ を満たすものが存在すると仮定する。 $|V|=2(N+M)+2$ なので、 G' には少なくとも2つ以上の連結成分が存在する。また補題2.1より、各連結成分は各領域内の節点を少なくとも一つずつ含む。以上のことから、 G' は二つの木 (T_1, T_0) とより構成される森でなければならない。全ての領域が二つの節点より構成されているので T_1, T_0 は各領域のそれぞれ異なる節点を一つずつ含む。

ここで、 $r_1 \in T_1, r_0 \in T_0$ と仮定して一般性を失わない。

T_1, T_0 の性質より、 $u_i \in T_1$ ならば $\neg u_i \in T_0$ であり、 $\neg u_i \in T_1$ ならば $u_i \in T_0$ である。よってSATにおいて

$u_i \in T_1$ ならば $u_i=1$ 、 $\neg u_i \in T_1$ ならば $u_i=0$

とすることができる。また、 $r_0 \in T_0$ より、任意の $j=1, 2, \dots, M$ に対して $D_j \in T_0$ である。よって任意の $j=1, 2, \dots, M$ に対し $C_j \in T_1$ である。

仮定より各 C_j と r_1 は T_1 の枝と節点を経由して連結でなければならないので、 C_j と接続しているある u_i または $\neg u_i$ は必ず T_1 に含まれている。よってSATにおいて前記のように変数 u_i の値を決めれば、各節 C_j の値は1となっている。よって、SATの問題例 (U, C) が充足可能、つまりyesであることが示された。

「SATの問題例 (U, C) がyes \rightarrow MNACの問題例 (G^a, h) がyes」の証明

(U, C) を充足する変数の値 $\{u_1^*, u_2^*, \dots, u_N^*\}$ が存在すると仮定する。

u_i^* に基づいて G^a 上の木 T_1, T_0 を次の様に構成する。 $r_1 \in T_1, r_0 \in T_0$ とする。各 $i=1, 2, \dots, N$ について、 $u_i^*=1$ ならば $u_i \in T_1, \neg u_i \in T_0$ 、 $u_i^*=0$ ならば $\neg u_i \in T_1, u_i \in T_0$ とする。また、すべての $j=1, 2, \dots, M$ に対して $D_j \in T_0$ とする。 (U, C) は u_i^* によって充足されているので、各節 C_j の値は1となる。すなわち、節 C_j に含まれるあるリテラル $(u_i$ または $\neg u_i)$ の値は1であるが、これは上の構成において C_j と接続しているある u_i または $\neg u_i$ が T_1 に含まれることを意味する。そこですべての $j=1, 2, \dots, M$ について $C_j \in T_1$ とする。この結果、2つの木 T_1 と T_0 が定まるので、これを $G'=(V, E')$ とする。構成法から容易に分かるように、 G' は G の全域部分グラフであり、その連結成分は T_1 と T_0 の二つであり、さらに T_1 と T_0 は全ての領域 V_i 上の点を夫々1つずつ含んでいるので、補題2.1より、 G' は G^a のNA連結な部分グラフである。また、 $|E'|=2(N+M)=h$ 。よって問題例 (G^a, h) はyesである。

(定理2.1の証明了)

定理2.1の証明で、SATの問題例を元に構築したMNACの問題例($G^*=(G,X),h$)において、常に G は連結であり、かつ全ての V_i について $|V_i|=2$ である。よってMNACにさらにこれらの制約を付与した問題もNP完全になる。この結果を以下に記す。

【定義】 2節点領域の全域独立2木問題(2 disjoint spanning Tree of 2-Nodes Area graph: 2T2NA)

[入力] 領域グラフ $G^*=(G,X)$ 、但し G は連結であり、かつ全ての領域は2節点より成る。

[要請] 2つの互いに節点を共有しない木 T_1, T_2 で、かつ任意の $V_i \in X$, $v, w \in V_i$ に対し、 $v \in T_1, w \in T_2$ である様な T_1, T_2 は存在するかどうかを判定せよ。□

【系2.1】 2T2NAはNP完全である。□

証明) 定理2.1の証明より明か。 Q.E.D.

通信網における設備の二重化問題が2T2NAに定式化され、近似アルゴリズムが提案されている[HAYA 92]。なお、文献[HAYA 92]においては本問題がNP完全であるか否かは未解決問題であった。

2. 2. 3. 最小NA連結問題の近似解法

定理2.1により、最小NA連結問題がNP完全であることが示された。しかし、本節では、この問題を枝数最小化を目的とした最適化問題と考えた場合、問題例 $G^*=(G,X)$ に対し絶対近似誤差が $\min(|V_i|)-K$ (但し K はの連結成分の数) で押さえられる線形時間の近似アルゴリズムが存在することを示す。この誤差の値はMNACOに対しては定数では無いが、前節の系2.1で示した様に、 $|V_i|=2$ 、 $K=1$ に限定したMNACである2T2NAもNP完全であり、この問題の最適化問題版にあてはめてみると、僅か $\min(|V_i|)-K=2-1=1$ の絶対近似誤差を保証する。

【定義】 最小 NA 連結問題 (MNACO; minimum NA-connected spanning subgraph problem) (最適化問題)

[入力] NA連結な領域グラフ $G^a=(G,X)$

[要請] 最小の $|E'|$ をもつ NA 連結全域部分グラフ $G'=(V,E')$ を求めよ。□

最小 NA 連結最適化問題は、定理 2.1 より当然 NP 困難である。しかし、 G が完全グラフの場合は次のような性質がある。

【定理 2.2】 G が完全グラフの時、 $G^a=(G,X)$, ($X=\{V_1, V_2, \dots, V_h\}$) の最小 NA 連結全域部分グラフ $G^*=(V,E^*)$ は、 $|E^*|=|V|-\min(|V_i|)$ を満たす。□

証明) まず $|E'|=|V|-\min(|V_i|)$ を満たす様に NA 連結全域部分グラフ G' を構成する。 $h=\min(|V_i|)$ とおく。各 V_i について空でない h 個の部分集合への分割を任意に作り $(V_i^1, V_i^2, \dots, V_i^h)$ とする。次に $V^j=V_1^j \cup V_2^j \cup \dots \cup V_h^j$, $j=1, 2, \dots, h$ と定め、部分グラフ $G_j=(V^j, E(V^j))$ の全域木を任意に選び T_j とする。 $G=T_1 \cup T_2 \cup \dots \cup T_h$ とすれば、各 T_j は全ての V_i について $T_j \cap V_i \neq \emptyset$ を満たすので、補題 2.1 より、 G' は G の NA 連結全域部分グラフである。また、 T_1, T_2, \dots, T_h は互いに共有節点を持たないので、 $|E'|=|T_1 \cup T_2 \cup \dots \cup T_h|=|V|-\min(|V_i|)$ である。よって $|E^*| \leq |V|-\min(|V_i|)$ 。

次に G^a の最小 NA 連結全域部分グラフ G^* が K 個の連結成分 $G^*_1, G^*_2, \dots, G^*_K$ より構成されているとする。補題 2.1 より、任意の V_i と任意の G^*_j について、 $V_i \cap V(G^*_j) \neq \emptyset$ でなければならない。よって $K \leq \min(|V_i|)$ 。さらに

$$|E^*| \geq |V| - K \geq |V| - \min(|V_i|).$$

上の二つの結果を合わせ、 $|E^*|=|V|-\min(|V_i|)$ を得る。 Q. E. D.

定理 2.2 は、完全グラフ G においては、節点数 $|V|$ と最小節点数領域の含む節点数 $\min(|V_i|)$ が決まれば、領域数や、他の領域の含む節点数等と関係なく、最小枝数の NA 連結グラフの枝数が決まることを示す。これは最小枝数の連結グラフの枝数が、 $|E|=|V|-1$ で与えられることの拡張である。

また、完全グラフに対する結果である定理 2.2 は、一般のグラフに対する最小 NA 連結最適化問題の下界値を与えているとも考えられる。よってこのことを利用して、近似アルゴリズムの誤差を評価することができる。一方、

MNACの実行可能解として次の定理に示す全域部分グラフが存在する。この正当性は補題2.1から簡単に導くことができる。

【定理2.3】 NA連結な領域グラフ $G^a=(G,X)$ に対し、 G の全域極大森を T とすると、 (T,X) もNA連結である。□

証明) 補題2.1より、 G の各連結成分は、すべての領域の少なくとも一節点を含む。 T は G の全域極大森なので、 G において同一連結成分に属する節点对は T においても同一連結成分に属する。よって、 T の各連結成分も、すべての領域の少なくとも一節点を含む。従って補題2.1より、 (T,X) はNA連結である。 Q.E.D.

グラフ $G=(V,E)$ の全域極大森は、適当な探索法を用いることにより、線形時間 $O(|V|+|E|)$ で見つけることができる [IBAR 89a]。よって定理2.2と定理2.3から以下の結果を得る。

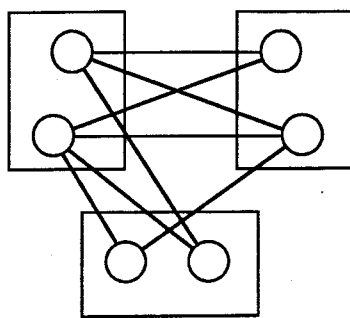
【定理2.4】 最小NA連結最適化問題(MNACO)には絶対近似誤差 $\min(|V_i|)-K$ の $O(|V|+|E|)$ 時間の近似アルゴリズムが存在する。ただし、 K は問題例 $G^a=(G,X)$ のグラフ G の連結成分の数である。□

証明) $G^a=(G,X)$ がNA連結であるとき、 G を張る全域極大森を T とすると、定理2.3より (T,X) もNA連結である。つまり T はMNACOの実行可能解である。 T の枝数は $m=|V|-K$ であり、MNACOの最適解 m^* は定理2.2より $m^* \geq |V| - \min(|V_i|)$ を満たすので、 $m - m^* \leq \min(|V_i|) - K$ が成立する。 G を張る全域極大森 T は $O(|V|+|E|)$ で見つけることができる。 Q. E. D.

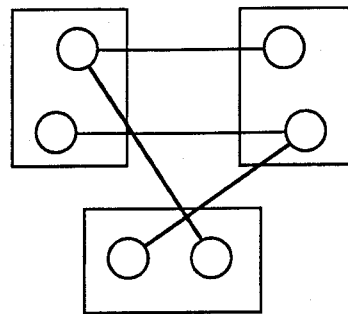
定理2.4より、領域グラフ $G^a=(G,X)$ が与えられたとき、 G の全域極大森を見つめることで、絶対近似誤差 $\min(|V_i|)-K$ 以下の近似解を得ることができる。この絶対近似誤差は、厳密には定数では無い。しかし、本定理は前述の様に、MNACの部分問題でNP完全である2T2NAに対して、絶対近似誤差1を保証する。実際、MNACOを $\min(|V_i|)$ と K の値を任意の自然数に固定した問題に分割しても、各問題はNP困難であり (但し $\min(|V_i|)=K=1$ の場合は、補題2.1より全域木が最適解となるので、除外する)、この近似アルゴリズム

ムはそれぞれのNP困難問題に対し、定数の絶対近似誤差を保証する。この意味で、定理2.4は安定した絶対近似誤差を保証すると言える。

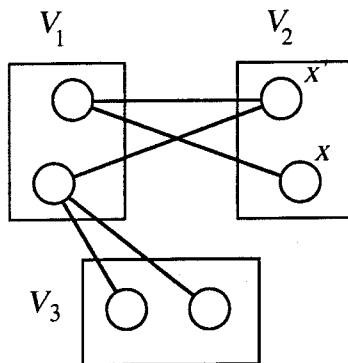
例えば、図2.5(a)の領域グラフ G^* のMNACOにおける最適（枝数最小）解は、例えば(b)によって与えられる（枝数は4）が、定理2.4によって(c)(d)の様な任意の全域木が、絶対近似誤差 $\min(|V_i|) - K = 2 - 1 = 1$ の近似解（枝数5）となる。



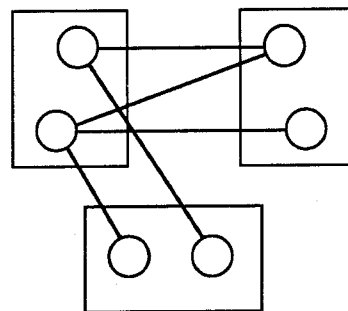
(a) 基のグラフ



(b) 枝数最小解



(c) 近似解その一



(d) 近似解その二

図2.5 最小NA連結問題とその近似解

Fig. 2.5. The optimum solution of MNACO and approximate solutions.

2. 3. 最小 k -NA連結問題

本章ではNA連結を拡張して、 k -NA連結という概念を導入する。領域を考慮した連結度については文献[HARA 69]等でも多少触れられているが、ここでは領域に隣接した節点は定義から除外している。ここでは、領域に隣接した節点との間にも定義を拡張する。

2. 3. 1. NA連結度とその性質

【定義】 グラフ $G=(V,E)$ において、 $V' \subset V$ と $x \in V-V'$ について、 $k'=|E(x,V')|$ としたとき、 $|W| < k-k'$ を満たす任意の $W \subseteq V-(V' \cup x)$ に対し、 G から、 V と W に隣接した枝全て、及び $E(x,V')$ を除いてできたグラフ $G''=(V'',E'')$ において、 x と V' がNA連結であるならば、 x と V' は k -NA連結 (k -NA connected) であると言う。また、 x と V' が k -NA連結である最大の k を x と V' のNA連結度 (NA-connectivity) とし、 $\kappa(x,V';G)=k$ と表現する。なお、誤解の怖れが無い場合には $\kappa(x,V')=k$ と書く。 $(x,y \in V)$ 間の連結度を表す記号 $\kappa(x,y;G)$ をそのまま用いた。

また、領域グラフ $G^a=(G,X)$ において、任意の領域 $V_i \in X$ と、 V_i に含まれない任意の節点 $x \in V-V_i$ に対して、 x と V_i が k -NA連結ならば、 G^a は k -NA連結であると言う。 G^a が k -NA連結である様な最大の k を G^a のNA連結度といい、 $\kappa(G^a)=k$ と表現する。

また、 k -NA連結な領域グラフ $G^a=(G,X)$ に対し、 $G=(V,E)$ の部分グラフ $G'=(V',E')$ と $X'=\{V'_i=V_i \cap V' \mid V_i \in X\}$ について、領域グラフ (G',X') が k -NA連結であるとき、 G' は G^a (あるいは G) の k -NA連結な部分グラフであるという。□

NA連結度の定義において、 $V_i=\{y\}$ と考えたものが、従来から定義されている (節点) 連結度の概念である。この意味でNA連結度は (節点) 連結度の拡張になっている。明らかに、1-NA連結とNA連結は等しい。 k -NA連結には、グラフの k -連結性における Menger の定理 [MENG 27] [HARA 69] から簡単に拡張できる、次の様な性質がある。

【定理2.5】 グラフ $G=(V,E)$ において、 $V' \subset V$ と $x \in V-V'$ が k -NA連結である必要十分条件は、 x と V' 間に k 本の内素な路が存在することである。□

本定理の証明の前に、 k -NA連結の別の必要十分条件と、 k 本の内素な路の存在に関する必要十分条件をそれぞれ補題として示しておく。

【補題2.2】 グラフ $G=(V,E)$ において、 $V' \subset V$ と $x \in V-V'$ が $|E(x, V')|=0$ を満たすとする。このとき、 V' と x が k -NA連結である必要十分条件は、 G の V' を一点 v' に縮約して得たグラフ $G/V'=(V/V', E/V')$ において v' と x が k -連結であることである。□

証明) $|E(x, V'; G)|=0$ であることから、 $|E(x, v'; G/V')|=0$ である。

(必要性の証明) 対偶を示す。 G/V' において v' と x が k -連結でないと仮定する。 $|E(x, v'; G/V')|=0$ であることを考慮すると、 k -連結性の定義より $|W| < k$ を満たすある $W \subseteq V/V' - \{x, v'\}$ が存在して、 v' と x は $G/V' - W$ において非連結である。すると、 V' と x は $G - W$ において非連結であり、 $W \subseteq V - V' - x$ であることを考慮すると、 G 上で考えても $|W| < k$ である。よって W の存在から、 G において V' と x は k -NA連結ではない。(必要性の証明了)

(十分性の証明) 対偶を示す。 G において V' と x が k -NA連結でないと仮定する。 $|E(x, V'; G)|=0$ であることを考慮すると、 k -NA連結性の定義より $|W| < k$ を満たすある $W \subseteq V - V' - x$ が存在して、 V' と x は $G - W$ においてNA連結でない。すると、 v' と x は $G/V' - W$ において非連結であり、 $W \subseteq V/V' - \{x, v'\}$ であることを考慮すると、 G/V' 上で考えても $|W| < k$ である。よって W の存在から、 G/V' において v' と x は k -連結ではない。(十分性の証明了) Q.E.D.

【補題2.3】 グラフ $G=(V,E)$ において、 $V' \subset V$ と $x \in V - V'$ が $|E(x, V')|=0$ を満たすとする。このとき、 V' と x 間に k 本の内素な路が存在する必要十分条件は、 G の V' を一点 v' に縮約して得たグラフ $G/V'=(V/V', E/V')$ において v' と x 間に k 本の内素な路が存在することである。□

証明) (必要性の証明) G において、 V' と x 間に k 本の内素な路が存在すると仮定する。その k 本の内素な路の集合を Q とする。 Q 内の任意の路を選んで、 $p = \langle x_1, x_2, \dots, x_{|p|}, x \rangle$ (但し $x_1, x_2, \dots, x_{|p|} \in V$) と表す。 $x_i, i=1, 2, \dots, |p|$ の

うち $x_i \in V'$ を満たす最大の i をもつ節点を x_i とする。(少なくとも $x_1 \in V'$ なので、このような x_i は必ず存在する。) このとき、 $p' = \langle x_i, x_{i+1}, \dots, x_{|p|}, x \rangle$ はやはり V' と x 間の路である。 Q の全ての路について同様の処理を加えて新たに路集合 Q' を作ると、 Q' は明かに k 本の内素な路の集合であり、しかも各路は V' の節点を丁度一つ (その端点として) 含む。次に Q' 内の任意の路 $p' = \langle x_i, x_{i+1}, \dots, x_{|p|}, x \rangle$ に対応して、 G/V' 上の路 $p'' = \langle v', x_{i+1}, \dots, x_{|p|}, x \rangle$ を作る。 Q' の全ての路について同様の処理を加えて G/V' 上の路集合 Q'' を作ると、 Q'' は明かに G/V' 上の k 本の内素な路の集合である。(必要性の証明了)

(十分性の証明) G/V' において v' と x 間に k 本の内素な路が存在すると仮定する。その k 本の内素な路の集合を Q とする。 Q 内の任意の路を選んで、 $p = \langle v', x_2, x_3, \dots, x_{|p|}, x \rangle$ (但し $x_2, x_3, \dots, x_{|p|} \in V$) と表す。当然 $e(v', x_2) \in E/V$ なので、 G/V' の作り方から、 G において $e(y, x_2) \in E$ を満たす $y \in V'$ が存在する。すると、 $p' = \langle y, x_2, x_3, \dots, x_{|p|}, x \rangle$ は G における V' と x 間の路であり、 Q の全ての路について同様の処理を加えて G 上の路集合 Q' を作ると、 Q' は明かに G 上の k 本の内素な路の集合である。(十分性の証明了) Q.E.D.

以上の補題から以下の様に定理2.5が証明される。

定理2.5の証明) $k' = |E(x, V')|$ とおく。 $k' \geq k$ ならば、 $k - k' \leq 0$ となるので、定義より x と V' は k -NA連結であり、かつ x と V' 間に k 本の内素な路が存在する。よって $k' < k$ の場合を示せば良い。

G において、 $E(x, V')$ は x と V' 間の k' 本の内素な路を与えることに注意して、 $E' = E - E(x, V')$ に基づいて $G' = (V, E')$ とする。すなわち、 G' において、 x と V が $(k - k')$ -NA連結である必要十分条件が、 x と V' 間に $k - k'$ 本の内素な路が存在することであることを以下に示す。

G' の V' を一点 v' に縮約して得られるグラフを $G'' = (V'', E'')$ とおく。すると補題2.2より、 G'' において、 x と v' が $k - k'$ 連結であることと G' において x と V が $(k - k')$ -NA連結であることは同値である。

ここで、 G'' において、 x と v' が $k - k'$ 連結である必要十分条件は、Mengerの定理によって x と v' 間に $k - k'$ 本の内素な路が存在することである。一方、補題2.3より G'' において、 x と v' 間に $k - k'$ 本の内素な路が存在することは、 G において、 x と V' 間に $k - k'$ 本の内素な路が存在することと同値である。

以上の結果、 G' において x と V が $(k - k')$ -NA連結である必要十分条件は、

G' において、 x と V' 間に k - k' 本の内素な路の存在することであることが示される。すなわち $G=(V,E)$ において、 $V'\subset V$ と $x\in V-V'$ が k -NA連結である必要十分条件は、 x と V' 間に k 本の内素な路が存在することである。 Q. E. D.

全ての節点と領域の組合わせについて k -NA連結性を調べれば、グラフの k -NA連結性は判定できる。節点と領域の組合わせについての k -NA連結性は、補題2.2で示した様に該当領域を縮約したグラフの k -連結性を調べれば分かる。以上から、与えられたグラフの k -NA連結性の判定を多項式時間で行なうアルゴリズムを得る。

```

procedure DECIDE
comment  $G^{*a}=(G,X)$ : 入力である領域グラフ。
begin
1  for each  $V_i\in X$  do
    begin
2       $V_i$ を一点 $v_i$ に縮約したグラフ $G/V_i=(V/V_i,E/V_i)$ を作る。
3      for each  $x\in V/V_i-v_i$  do
        begin
4           $\kappa(x,v_i;G/V_i)\geq k$ か否かを判定する。
5          if  $\kappa(x,v_i;G/V_i)<k$  then "no"を出力 stop;
        end;
    end;
6  "yes"を出力
end;
```

【補題2.4】 手続きDECIDEは、与えられた領域グラフ $G^{*a}=(G=(V,E),X)$ が k -NA連結であるか否かを、 $O(|X||V|f(|V|,|E|))$ 時間で判定するアルゴリズムである。但し、 $f(|V|,|E|)$ は、多重グラフ $G=(V,E)$ の特定の節点間の k -連結性を判定するのに要する計算量を表す。□

証明) 手続きDECIDEが、「与えられた領域グラフが k -NA連結であるか否かを判定する」アルゴリズムであることは、補題2.2とグラフの k -NA連結性の定義より明か。次に計算量を見積る。手続きの第1行のdo文は $|X|$ 回繰

り返される。第2行の操作は一回につき $O(|V|+|E|)$ で可能である。よってアルゴリズム全体では、第2行の操作は $O(|X|(|V|+|E|))$ でできる。第1行のdo文を一回実行する毎に第3行のdo文は $|V|$ 回繰り返される。第4行の判定は一回につき $f(|V|, |E|)$ 時間でできる（一般に G/V_i は多重グラフになる）ので、アルゴリズム全体では、第4行の判定は $O(|X||V|f(|V|, |E|))$ 時間でできる。その他の操作の計算量はこれらで押さえられる。

以上から、このアルゴリズムの計算量は

$$O(|X||V|f(|V|, |E|) + |X|(|V|+|E|)) = O(|X||V|f(|V|, |E|))$$

となる。 Q.E.D.

$G=(V, E)$ の特定の節点間の k -連結性の判定に関しては、 $O(\min(k, |E|^{1/2})|E|)$ でできるアルゴリズムが知られている [EVEN 75] [NAGA 93]。このことから直ちに次の定理を得る。

【定理2.6】 与えられた領域グラフが k -NA連結であるか否かは多項式時間（高々 $O(\min(k, |E|^{1/2})|X||V||E|)$ ）で判定できる。□

証明）補題2.4と、 $G=(V, E)$ の特定の節点間の k -連結性を判定する、 $O(\min(k, |E|^{1/2})|E|)$ 時間アルゴリズムの存在より明か。 Q.E.D.

手続き DECIDE の一部を変更するだけで、与えられたグラフの NA 連結度を計算するアルゴリズムを得る。第4行の $\kappa(x, v_i; G/V_i) \geq k$ の判定の代わりに $\kappa(x, v_i; G/V_i)$ を計算し、その最小値を記憶しておけば良い。全ての $V_i \in X$, $x \in V - V_i$ の組合わせについて計算して、その最小値を出力すれば G^* の NA 連結度を得る。このアルゴリズムの計算量は、手続き DESIDE の場合と同様の考察により、多重グラフ $G=(V, E)$ の特定の節点間の連結度の計算量を $g(|V|, |E|)$ とおくと、 $O(|X||V|g(|V|, |E|))$ となる。多重グラフ $G=(V, E)$ の特定の節点間の連結度は、 $O(|V|+|E|^{3/2})$ で計算できることが知られている [EVEN 75] [NAGA 93]。このことから与えられたグラフの NA 連結度は、多項式時間（高々 $O(|X||V|(|V|+|E|^{3/2}))$ ）で計算できる。よって次の定理を得る。

【定理2.7】 与えられた領域グラフの NA 連結度は多項式時間（高々 $O(|X||V|(|V|+|E|^{3/2}))$ ）で計算できる。□

証明) 上記の議論から明か。 Q.E.D.

2. 3. 2. k -連結性と k -NA連結性の関係

k -NA連結性に対応して、次の最小化問題を定義する。

【定義】 最小 k -NA連結問題 (Minimum k -NA-Connected subgraph problem: MknAC) (決定問題)

[入力] k -NA連結な領域グラフ $G^a=(G,X)$, $h \in \{\text{自然数}\}$ 。

[要請] $|E'| \leq h$ である様な k -NA連結全域部分グラフ $G'=(V,E')$ が存在するかどうかを判定せよ。□

最小 k -NA連結問題 MknAC は最小 NA連結問題を部分問題として含むので、定理2.1より NP困難である。与えられた領域グラフが k -NA連結であることの判定は定理2.6より多項式時間で可能であることを考慮すれば、MknAC は NPに属する、すなわち NP完全である。また、本問題はグラフの最小 k -連結問題 (k -NA連結を k -連結に置き換えた問題) の拡張とも考えられるが、この最小 k -連結問題も既に NP完全である (ハミルトン閉路問題[GARE 79]を含む)。よって MknAC の最適化問題版を考えて、近似アルゴリズムの可能性を検討する。

【定義】 最小 k -NA連結問題 (Minimum k -NA-Connected subgraph Optimization problem: MknACO) (最適化問題)

[入力] k -NA連結な領域グラフ $G^a=(G,X)$ 。

[要請] 最小の $|E'|$ をもつ、 G の k -NA連結全域部分グラフ $G'=(V,E')$ を求めよ。□

最小 k -連結最適化問題には絶対近似誤差が定数で押さえられる近似アルゴリズムの存在は知られていない。しかし、次の様な性質を利用した、相対近似誤差が定数で押さえられる近似アルゴリズムが知られている[NAGA 92]。

【定義】 グラフ $G=(V,E)$ に対し、全域部分グラフ $G'=(V,E')$ で、任意の節点对 $x,y \in V$ について $\kappa(x,y;G') \geq \min(k, \kappa(x,y;G))$ を満たすものを、 k -連結性を保存する部分グラフと呼ぶ。また k -連結性を保存する部分グラフで枝数が $k|V|$ 以下のものを k -連結性を保存する疎な部分グラフと呼ぶ。□

【定理2.A】 [NAGA 92] 任意のグラフ $G=(V,E)$ 、任意の整数 k に対し、 k -連結性を保存する疎な部分グラフ $G'=(V,E')$ が存在し、そのうちの一つを $O(|V|+|E|)$ 時間で見つけることができる。□

グラフが k -連結であるためには、各節点に k 本以上の枝が接続している必要があるので、最小 k -連結最適化問題の解の下限は $\lceil k|V|/2 \rceil$ である。故に、定理2.Aで得られる G' は、相対近似誤差 2 の近似解である。

k -NA連結に対しても、以下に示す様にこの様な都合の良い部分グラフが存在する。

【定義】 領域グラフ $G^*=(G=(V,E),X)$ と整数 k に対し、 G の全域部分グラフ $G'=(V,E')$ で、任意の $V_i \in X$ と任意の $x \in V - V_i$ について

$$\kappa(x, V_i; (G', X)) \geq \min(k, \kappa(x, V_i; (G, X)))$$

を満たすものを、 k -NA連結性を保存する部分グラフと呼ぶ。また k -NA連結性を保存する部分グラフで枝数が $k|V|$ 以下なものを k -NA連結性を保存する疎な部分グラフと呼ぶ。□

次定理は、連結度を保存すればNA連結度も保存されることを示す。

【定理2.8】 領域グラフ $G^*=(G,X)$ に対し、グラフ G の k -連結性を保存する任意の部分グラフ G' は、 k -NA連結性を保存する。□

定理の証明の準備として、まず次の補題を示す。

【補題2.5】 グラフ $G=(V,E)$ に対し、 k -連結性を保存する任意の部分グラフを $G'=(V,E')$ とする。このとき $e(x,y) \in E$ かつ $\kappa(x,y;G) \leq k$ ならば $e(x,y) \in E'$ である。□

証明) 題意が成立しないと仮定する。すなわち、
ある $e(x,y) \in E - E'$ が存在して、 $\kappa(x,y;G) \leq k$ が成立すると仮定する。
 $\kappa(x,y;G) \leq k$ より $\kappa(x,y;G') = \kappa(x,y;G)$ 。 $E'' = E' \cup \{e(x,y)\}$, $G'' = (V, E'')$ とする。
明らかに $\kappa(x,y;G'') = \kappa(x,y;G') + 1$ 。
また $E'' \subseteq E$ より $\kappa(x,y;G'') \leq \kappa(x,y;G)$ 。
よって $\kappa(x,y;G) \geq \kappa(x,y;G'') > \kappa(x,y;G') = \kappa(x,y;G)$ となり矛盾。 Q. E. D.

定理2.8の証明) 題意が成立しないと仮定する。つまり、ある領域グラフ $G^{*a} = (G = (V, E), X)$ の k -連結性を保存する部分グラフ $G' = (V, E')$ に対し、
 $\kappa(x, V_i; G^{*a}) = k_1 \leq k$ かつ $\kappa(x, V_i; (G', X)) = k_1' < k_1$ を満たす $V_i \in X$ と $x \in V - V_i$ が存在すると仮定する。

明らかに、任意の $y \in V_i$ に対し $\kappa(x, y; G) \leq \kappa(x, V_i; G^{*a}) = k_1 \leq k$ 。

よって補題2.5より $E(x, V_i) \subseteq E'$ 。

$k_2 = |E(x, V_i)|$ とおく。

$E(x, V_i) \subseteq E'$ より $k_2 \leq k_1' < k_1$ 。

そこで、 $F = E - E(x, V_i)$, $G_f = (V, F)$, $F = E' - E(x, V_i)$, $G_f' = (V, F')$ とすると、

$\kappa(x, V_i; (G_f', X)) = k_1' - k_2$ 。

よって $|W| = k_1' - k_2$ かつ $(x \cup V_i) \cap W = \emptyset$ を満たす $W \subseteq V$ で、 G_f' において x と V_i を分離するものが存在する。そこで $G_f' - W$ において x と同じ連結成分に属する節点集合を V^x 、それ以外の節点集合を V^y とおく ($V^y = V - W - V^x$ 、図2.6参照)。

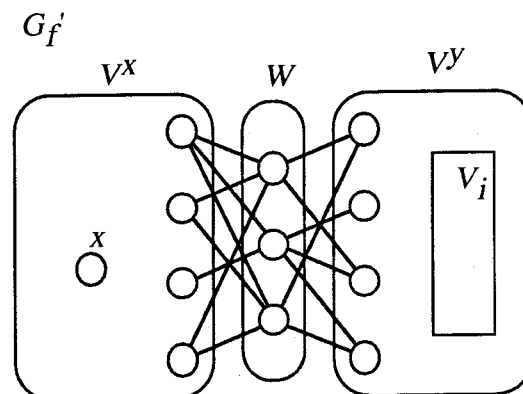


図2.6. V^x, V^y, W の説明図

Fig. 2.6. Explanation of V^x, V^y and W .

$\kappa(x, V_i; (G_f, X)) = k_1 - k_2$ より、 G_f 上に x と V_i の間に $k_1 - k_2$ 本の内素な路 $p_1, p_2, \dots, p_{k_1 - k_2}$ が存在する。さらに $k_1 - k_2 > k_1' - k_2$ より、 $p_1, p_2, \dots, p_{k_1 - k_2}$ のうち、少なくとも一つは W に含まれる節点を全く経由しない。その路を p_1 とする（一般性は失わない）。

p_1 上の節点を x 側から順に $x_0, x_1, x_2, \dots, x_v$ （但し $x = x_0, x_v \in V_i$ ）と記す。
 $x_0 \in V^x, x_v \in V^y$ なので p_1 上のある枝 $e(x_j, x_{j+1})$ について、 $x_j \in V^x, x_{j+1} \in V^y$ が成立する（図2.7参照）。このとき、 $G_f' - W$ において、 x_j と x_{j+1} は異なる連結成分に属する。故に $\kappa(x_j, x_{j+1}; G_f') \leq k_1' - k_2$ 。

よって $\kappa(x_j, x_{j+1}; G') \leq k_1' < k_1 \leq k$ 。

G' は k -連結性を保存するので、 $\kappa(x_j, x_{j+1}; G) \leq k_1' < k$ 。

よって補題2.5より $e(x_j, x_{j+1}) \in E$ でなければならないが、
 $e(x_j, x_{j+1}) \in E' - E(x, V_i)$ かつ $x_j, x_{j+1} \in V - W$ なので、これは $G_f' - W$ において、 x_j と x_{j+1} が異なる連結成分に属することと矛盾する。 Q. E. D.

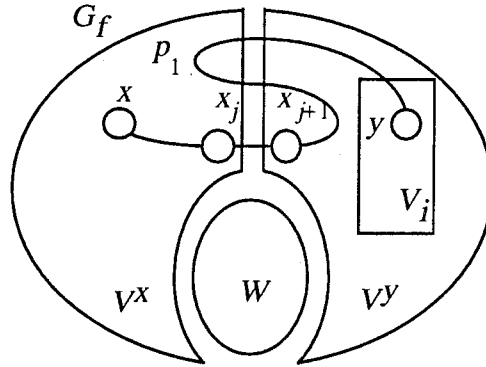


図2.7. p_1, x_j, x_{j+1} の説明図

Fig. 2.7. Explanation of p_1, x_j and x_{j+1} .

定理2.8によって、 k -連結性を保存する疎な部分グラフを作成することが、そのまま k -NA連結性を保存する疎な部分グラフを作ることになる。すなわち、定理2.Aで記した文献[NAGA 92]の様な高速アルゴリズムをそのまま用いることで、 k -NA連結性を保存する疎な部分グラフを得る。このことから次の定理を得る。

【定理2.9】 最小 k -NA連結最適化問題(MkNACO)には、相対近似誤差が2で押さえられる $O(|V|+|E|)$ 時間近似アルゴリズムが存在する。□

証明) 定理2.Aより、任意のグラフ $G=(V,E)$ と任意の整数 k に対し、 k -連結性を保存し、 $|E'| \leq k|V|$ を満たす部分グラフ $G'=(V,E')$ を、 $O(|V|+|E|)$ 時間で見つけることができる。定理2.8より G' は k -NA連結性も保存する、すなわち、領域グラフ (G,X) が k -NA連結であるならば、 (G',X) も k -NA連結である。よって G' はMkNACOの実行可能解である。一方、領域グラフが k -NA連結であるためには、各節点に k 本以上の枝が接続している必要があるので、MkNACOの解の下限は $\lceil k|V|/2 \rceil$ である。故に、 G' の、相対近似誤差は高々

$$k|V| / \lceil k|V|/2 \rceil \leq 2$$

である。故に、定理2.AのアルゴリズムはMkNACOに対する相対近似誤差が2で押さえられる線形時間近似アルゴリズムでもある。 Q.E.D.

定理2.9によって、「複数の交換機が集まって論理的な領域を構成している通信網の強度（NA連結度）を保証しながら、網を簡素にする問題」のモデル化である、最小 k -NA連結最適化問題(MkNACO)に対し、高速に近似解を得ることができる。定理2.8は近似アルゴリズムに依存せず成立するので、[NAGA 92]のアルゴリズム以外でも、 k -連結性を保存する部分グラフを作るアルゴリズムならば、MkNACOに適用できる。

2. 4. T -混合カット上のNA連結

前節までで扱ったNA連結度は、点連結度を拡張した概念であった。連結度には点連結度の他に枝連結度もある。通信網上で考えれば、点連結度は交換機（あるいは局）の故障に対する網の耐久力を表すのに対し、枝連結度は回線故障に対する網の耐久力を表し、どちらの概念も応用上重要な意味を持つ。よって前節までで扱った、領域グラフとNA連結度に関する諸定理も、点連結度だけでなく、枝連結度に関しても成立することが望ましい。

点連結度と枝連結度の両者を含んだ一般化概念として、 T -混合カットに基づく連結度[BEIN 67] [FRAN 93]がある。本節では、前節までに示した性質が、この概念に対して成立することを示す。このことは、当然、枝連結度に関しても成立することを意味する。

【定義】 グラフ $G=(V,E)$ と $T \subseteq V$ が与えられている。 $x, y \in V$ に対し、 x, y 間の路の集合 P は、次の性質(1), (2)を満たすとき、 T -独立(T -independent)と呼ぶ。

(1) P 内の路は互いに枝素である。

(2) T - x - y のどの節点もそれを通る P の路が高々1本しか無い。

T -独立な路の最大本数を $\lambda_T(x, y; G)$ で表す。 G が明らかな時は $\lambda_T(x, y)$ とも記す。□

【定義】 グラフ $G=(V,E)$ と $T \subseteq V$ が与えられている。 $Z \cup A \cup B = V$, $Z \cap A = Z \cap B = A \cap B = \phi$, $Z \subseteq T$, $A \neq \phi$, $B \neq \phi$ である時、 $C=(Z, A, B)$ を T -混合カットと呼び、カットの大きさを $|C| := |E(A, B)| + |Z|$ で定義する。また、 T -混合カット $C=(Z, A, B)$ は $x \in A$ と $y \in B$ を分離すると言う。□

V -独立は、内素と同値であり、 ϕ -独立は、枝素と同値である。また、2節点 $x, y \in V$ を分離する V -混合カットの最小値は、点連結度 $\kappa(x, y)$ と同値であり、2節点 $x, y \in V$ を分離する ϕ -混合カットの最小値は、枝連結度 $\lambda(x, y)$ と同値である。すなわち、 T -独立（ T -混合カットの最小値）は内素と枝素（点連結度と枝連結度）の一般化概念である。 $\lambda_T(x, y)$ と T -混合カットの間には次の形でMengerの定理が成り立つ。

【定理2.B】 [FRAN 93] [NAGA 93] $\lambda_T(x,y)$ は x と y を分離する T -混合カットの大きさの最小値に等しい。□

定理2.Aで述べた[NAGA 92]の線形時間アルゴリズムは $\lambda_T(x,y)$ に関しても、 k -連結性を保存することが分かっている[FRAN 93]。すなわち、以下の定理が存在する。

【定理2.C】 [FRAN 93]任意のグラフ $G=(V,E)$ と $T \subseteq V$ と任意の自然数 k に対し、 G の全域部分グラフ $G'=(V,E')$ で、

$$[\lambda_T(x,y;G') \geq \min(\lambda_T(x,y;G), k), \forall x,y \in V] \text{ かつ } [|E'| \leq k|V|]$$

を満たすものが存在し、そのうちの一つを $O(|V|+|E|)$ 時間で見つけるアルゴリズムが存在する。□

本節では、前節の定理2.8の結果を拡張して、定理2.Cの結果を、さらにNA連結度を拡張した概念にも適用できる様にする。定理2.8は、領域の与え方に関係なく成立したことに注意すると、任意の節点部分集合間に連結度を拡張しても、その連結度に対して定理2.8は成立する。そこで、NA連結度に対し、

(1) T -混合カットに関する連結度への拡張

(2) 任意の節点部分集合間の連結度への拡張

の2つを同時に行なっても、なお定理2.8は成立することを、以下に示す。

(2)の拡張を行なうために、 T -混合カットの概念を以下の様に拡張する。

【定義】 グラフ $G=(V,E)$ と $T \subseteq V$ が与えられている。 $X, Y \subset V$ に対し、 X, Y 間の路の集合 P は、次の性質(1), (2)を満たすとき、 T -独立と呼ぶ。

(1) P 内の路は互いに枝素である。

(2) T - X - Y のどの節点もそれを通る P の路が高々1本しか無い。

T -独立な路の最大本数を $\lambda_T(X,Y;G)$ で表す。 G が明らかな時は $\lambda_T(X,Y)$ とも記す。また、 T -混合カット $C=(Z,A,B)$ は $X \subseteq A$ と $Y \subseteq B$ を分離すると言う。

□

前節までに扱ったNA連結度 $\kappa(x,V)$ は、 T -独立な路の最大本数を用いた表

現をすれば、 $\lambda_T(x, V)$ になる。次に、 T -独立な路の最大本数 $\lambda_T(X, Y; G)$ に関しても、定理2.8と同様の性質が成立することを示す。

【定理2.10】 グラフ $G=(V, E)$ と $T \subseteq V$ と任意の自然数 k が与えられている。 G の全域部分グラフ $G'=(V, E')$ が、任意の節点对 $x, y \in V$ に対して

$$\lambda_T(x, y; G') \geq \min(\lambda_T(x, y; G), k)$$

を満たすならば、任意の節点部分集合対 $X, Y \subseteq V$ (但し $X \cap Y = \phi$) に対し、

$$\lambda_T(X, Y; G') \geq \min(\lambda_T(X, Y; G), k)$$

が成立する。□

本定理と定理2.8との関係は、定理2.10において、 $T=V$ に限定し、かつ $X=\{x\}$ ($x \in V-V_i$), $Y=V_i \in X$ に限定したものが、定理2.8になる。定理2.10の証明は定理2.8と同じ方針で出来るので省略する。本結果を前述の[FRAN 93]の結果と合わせると、以下の定理を得る。

【定理2.11】 任意の節点对 $x, y \in V$ に対して、 $\lambda_T(x, y; G) \geq k$ であるグラフ $G=(V, E)$ と $T \subseteq V$ が与えられている。このとき、 G の全域部分グラフ $G'=(V, E')$ で、「 $\lambda_T(x, y; G') \geq k, \forall x, y \in V$ 」かつ「 $|E'| \leq k|V|$ 」である様なものが存在し、そのうちの一つを $O(|V|+|E|)$ 時間で見つけることができる。
□

証明) 定理2.Cと定理2.10より明か。 Q.E.D.

T -混合カットは、通信網上では、枝故障と交換機（あるいは局）故障を同時に考慮する問題に応用できる。 $T \subseteq V$ を適当に与えることにより、故障を考慮しない交換機と考慮する交換機とを分けて考えることができる。本節の結果により、こういった、 T -混合カットに定式化される問題も巧く解けるようになる。

2. 5. 最小直NA連結問題

本節では、NA連結にさらに「発節点と同じ領域内の節点を経由しない経路が存在する。」という条件を加えた制約（直NA連結と呼ぶことにする。正確な定義は本文内で行なう。）下の問題を考える。本制約の通信網上で意味は、領域が一つの県に対応しているとすると、地震等の大規模災害において、県全体に被害が及んだ時に、一つでも交換機が生き残っていれば、そこから他の全ての場所に通信が可能となることを意味する。また、発信領域内の各節点の処理する通信量を押さえる効果と、管理者が通信の流れを把握し易くするという効果もある。

2. 2で示したように、与えられた領域グラフに対し、NA連結である最小枝数全域部分グラフを求める問題はNP困難であるが、任意の全域木はNA連結性を保つという都合の良い性質があった。しかし本節では、直NA連結性を満たす全域木を求める問題事態がNP困難になることを示す。さらに、ある簡単な条件の下で、任意の領域グラフに対し、枝数 $2|V|-3$ 以下の直NA連結全域部分グラフを求める時間計算量 $O(|E||X|)$ の近似アルゴリズムが存在することを示す。

2. 5. 1. 直NA連結性

【定義】 領域グラフ $G^a=(G=(V,E),X)$ の、 $V_i \in X$ と $x \in V-V_i$ について、 x を含む領域を V' とすると、 x と V_i 間の路で $V'-x$ の節点を経由しないものが存在するとき、 x と V_i は直NA連結(direct NA-connected)であると言う。また、任意の $V_i \in X$ と任意の $x \in V-V_i$ について、 x と V_i が直NA連結である時、 G^a は直NA連結であると言う。

また、直NA連結な領域グラフ $G^a=(G,X)$ に対し、 $G=(V,E)$ の部分グラフ $G'=(V',E')$ と $X'=\{V'_i=V_i \cap V' \mid V_i \in X\}$ について、領域グラフ (G',X') が直NA連結であるとき、 G' は G^a (あるいは G) の直NA連結な部分グラフであるという。□

図2.5(a)のグラフは直NA連結であり、(b),(d)もそうである。しかし(c)は x から V_3 に行くのに x' を通らねばならず、直ではない。

まず、領域グラフ G^a が直NA連結であるための条件を以下の補題として掲げておく。

【補題2.6】 領域グラフ $G^a=(G=(V,E),X)$ において、領域 $V_i \in X$ と節点 $x \in V_j$ ($i \neq j$) が直NA連結である必要十分条件は、 $G-(V_j-x)$ における、 x を含む連結成分が、 V_i 内の節点を少なくとも一つ含むことである。□

証明) (必要性の証明) 領域グラフ $G^a=(G=(V,E),X)$ が直NA連結であると仮定する。すると、定義から、任意の節点 $x \in V_j$ と、 x を含まない任意の領域 $V_i \in X$ ($i \neq j$) 対して、 x, V_i 間に V_j-x を通らない路 $p(x, V_i)$ が存在する。よって $E(p(x, V_i)) \subseteq E(G-(V_j-x))$ となる。これは、 $G-(V_j-x)$ において、 x とある V_i 内のある節点が連結であることを表す。すなわち、 $G-(V_j-x)$ における、 x を含む連結成分が、 V_i 内の節点を少なくとも一つ含む。(必要性の証明了)

(十分性の証明) 領域グラフ $G^a=(G=(V,E),X)$ の任意の節点 $x \in V_j$ と、 x を含まない任意の領域 $V_i \in X$ ($i \neq j$) に対し、 $G-(V_j-x)$ における、 x を含む連結成分が、 V_i 内の節点を少なくとも一つ含むと仮定する。すると、 $G-(V_j-x)$ において x と V_i 間に路 $p(x, V_i)$ が存在する。 $p(x, V_i)$ は明かに V_j-x を通らない。よって G^a は直NA連結である。(十分性の証明了) Q.E.D.

補題2.6より次の条件も導ける。

【補題2.7】 領域数 $|X|$ が2以上である領域グラフ $G^a=(G=(V,E),X)$ が直NA連結であるならば、任意の節点 $x \in V$, x を含む領域 $V_j \in X$ について、 $E(x, V-x) \cap E(V_j, V-V_j) = \emptyset$ である。□

証明) 領域数が2以上である領域グラフ $G^a=(G=(V,E),X)$ において、ある節点 $x \in V$, x を含む領域 $V_j \in X$ について、 $E(x, V-x) \cap E(V_j, V-V_j) = \emptyset$ であると仮定する。つまり、 x に接続する任意の枝は V_j 内の節点との間の枝である。すると $G-(V_j-x)$ における x を含む連結成分は、 $\{x\}$ のみとなる。領域数が2以上であることから、 V_j 以外の領域 $V_i \in X$ ($i \neq j$) が存在し、 V_i 内の節点は、 x を含む連結成分に含まれない。よって補題2.6より G^a は直NA連結ではない。

Q.E.D.

2. 5. 2. 直NA連結性の判定

本節では、与えられた領域グラフの直NA連結性の判定を高速に行なうアルゴリズムを考える。

【定理2.12】 領域グラフ $G^{ra}=(G=(V,E),X)$ の直NA連結性は、 $O(|E||X|+|V|^2)$ で判定できる。□

証明) 領域 $V_i \in X$ と節点 $x \in V - V_i$ を与えたとき、 V_i と x がNA連結であるか否かは、 x を含む領域を V_j とすると、補題2.6より、 x を含む $G-(V_j-x)$ の連結成分が、 V_i 内の節点を少なくとも一つ含むかどうかを判定すれば良い。任意の $V_i \in X$ と $x \in V - V_i$ について、この判定を効率的に行なうアルゴリズムを以下に説明する。

```
procedure DECIDE-D
comment  $G^{ra}=(G=(V,E),X)$ : 入力 of 直NA連結領域グラフ
begin
1   $Y:=X$ 
2  while  $Y \neq \phi$  do
    begin
3       $Y$  に属する領域  $V_j$  を選ぶ;  $Y:=Y-\{V_j\}$ 
4       $Z:=V_j$ 
5       $G-V_j$  の連結成分  $G_j^k=(V_j^k, E_j^k)$ ,  $k=1, 2, \dots, K$  を作成する (但し  $K$  は
         $G-V_j$  の連結成分の数)。
6      while  $Z \neq \phi$  do
          begin
7               $Z$  に属する節点  $x$  を選ぶ;  $Z:=Z-\{x\}$ 
8               $x$  に隣接する全ての節点  $y \in V - V_j$  に対し、 $y$  を含む連結成分  $G_j^k$  の
                節点集合  $V_j^k$  を合わせて、節点集合  $V_j(x)$  を作成する。
9              if ある  $V_i \in X - \{V_j\}$  内の節点が、 $V_j(x)$  内に一つも存在しない
10             then "no" を出力 stop;
          end;
    end;
end;
```

11 "yes" を出力

end;

前述の様に、 G^{*a} の直NA連結性の判定は、任意の $V_i \in X$ と $x \in V - V_i$ について x を含む $G - (V_i - x)$ (但し $x \in V_i$) の連結成分が、 V_i 内の節点を少なくとも一つ含むかどうかを判定すれば良く、第9行の判定がそれと等しいので、手続きDECIDE-Dが G^{*a} の直NA連結性を正しく判定することが分かる。

次にDECIDE-Dの計算量を見積る。

入力のグラフのデータは、各節点に接続する枝のデータをリストで作成してあるものとする。第2行の繰り返しは $|X|$ 回行なわれる。第5行の連結成分の作成は $O(|V|+|E|)$ 時間で行なえる。その際に各 V_i^k 毎に、 V_i^k に含まれる節点のリスト L_i^k と、各節点が属する連結成分番号の配列 D を作成しておく。第6行は各節点について一度ずつ行なわれるので、アルゴリズム全体で $|V|$ 回行なわれる。第8行の $V_j(x)$ の作成は、 x の隣接点 y を列挙 (アルゴリズム全体で $O(|E|)$ 時間) し、配列 D によって y の含まれる連結成分を求め、そのリスト L_i^k を接続することで、 $V_j(x)$ に含まれる節点リスト $L_j(x)$ として求めることが出来る。この部分はアルゴリズム全体で $O(|E|)$ 時間。第9行の判定は、 $L_j(x)$ に含まれる節点を含む領域を列挙していくことで可能で、一つの x につき $O(|V|)$ 時間でできる。以上から、アルゴリズム全体の計算量は

$$O(|X|(|V|+|E|)+|E|+|V|^2) \leq O(|E||X|+|V|^2)$$

である。(定理2.12の証明了)

2. 5. 3. 最小直NA連結問題

次に最小NA連結問題に直制約を加えた問題、最小直NA連結問題(MDNAC)を考える。

【定義】 最小直NA連結問題(Minimum Direct NA-Connected subgraph problem: MDNAC)

[入力] 直NA連結な領域グラフ $G^{*a}=(G,X)$, $h \in \{\text{自然数}\}$

[要請] $|E'| \leq h$ である様な直NA連結全域部分グラフ $G'=(V,E')$ は存在するかどうかを判定せよ。□

【定理2.13】 MDNACはNP完全である。□

証明) 定理2.1の証明と同じ方針でできる。違う点は、

- (1) SATを帰着した領域グラフが直NA連結になっていること、
- (2) 解は直NA連結になっていること、

の2点を示せば良い。

(1)の証明) 記号は定理2.1の証明の記号をそのまま用いる。

r_0 は全ての領域と隣接している（領域内のある節点に隣接している）ので、 r_0 の任意の隣接節点と任意の領域は直NA連結である。

r_0 と隣接していない節点は r_1 と $C_j, j=1,2,\dots,M$ である。

C_j とある $V_i, i=1,2,\dots,N$ は隣接しているので、 C_j から V_i を介して r_0 へ長さ2の路が存在する。よって C_j と任意の領域は直NA連結である。

r_1 と任意の $V_i, i=1,2,\dots,N$ は直NA連結である。のこるは r_1 と $V_{N+j}, j=1,2,\dots,M$ だが、 $C_j \in V_{N+j}$ が、ある $V_i, i=1,2,\dots,N$ に隣接しているので、 r_1 から V_i を介して V_{N+j} へ長さ2の経路が存在することになる。よって r_1 と $V_{N+j}, j=1,2,\dots,M$ は直NA連結である。

(2)の証明) 解は2つの節点を共有しない T_1, T_0 であり、各木は各領域の節点の一つずつ含む。よって解は明らかに直NA連結である。 Q.E.D.

この様にMDNACはNP完全であるので、最小NA連結問題(MNAC)と最小 k -NA連結問題(MkNAC)で行なった様に、最適化問題版 (MDNACOと呼ぶ) を考えた上で、枝数に余裕を持たせることで、MNACと同様に近似解を求めることを考える。

MNACに関しては、 G の全域木もNA連結であるという性質があった (定理2.3) ので、一つの全域木を作成することで絶対近似誤差 $\min(|V_i|)-K$ 以内の近似解が得られた (但し K は問題例のグラフの連結成分の数)。ところが、直NA連結の場合は図2.5(c)の例が示す様に単純に全域木とったのでは上手くいかない。この点について次節でさらに検討する。

2. 5. 4. 直NA連結全域木問題のNP完全性

【定義】 直NA連結全域木問題(DNACT)

[入力] 直NA連結な領域グラフ $G^a=(G,X)$

[要請] 直NA連結である全域木 T は存在するかどうかを判定せよ。□

定義より、グラフが直NA連結であるならばNA連結なので、MNACOの解の下界値 $|V| - \min(|V_i|)$ (定理2.2) は、MDNACOの下界値でもある。よって、DNACTが効率的に解ければ、MDNACの有力な近似アルゴリズムとなり得る。しかし、これを否定する結果が得られた。

【定理2.14】 DNACTはNP完全である。□

本定理を証明する為に、2. 2. 1で定義したNP完全問題、2 節点領域の全域独立木問題(2T2NA)を用いる。2T2NAの入力を直NA連結である領域グラフに限った問題を2T2NA'とする。2T2NAがMNACの部分問題であり、MNACのNP完全性の証明 (定理2.1) をそのまま2T2NAのNP完全性の証明 (系2.1) に使った。これと同様に、2T2NA'はMDNACの部分問題であり、MDNACのNP完全性の証明 (定理2.13) をそのまま使って、2T2NA'のNP完全性を証明できる。この結果を次に系として示す。

【系2.2】 2T2NA'はNP完全である。□

証明) 定理2.13の証明より明か。 Q.E.D.

定理2.14の証明)

全ての領域 $V_i \subset X$ が2 節点よりなる、任意の領域グラフ $G^a=(G=(V,E),X)$ に対し、2T2NA'の解の存在とDNACTの解の存在が一致する (片方の解からもう片方の解が作成できる) ことを示す。但し G は連結であると仮定して良い。なぜなら、非連結かつNA連結ならば、領域の節点数が2 なので、定理2.4で示した近似アルゴリズムによって得られた近似解 (極大全域森) の近似誤差が

$$\min(V_i) - K = 2 - 2 = 0$$

となる、つまり、最適解となっている。よって、非連結グラフを入力から除外しても2T2NA'のNP完全性は変わらないからである。

(1) 2T2NA'の解からDNACTの解を作る方法。

2T2NA'の解 G' は枝数 $|V|/2-1$ である二つの木 T_1, T_2 から構成される森である。 $T_1(T_2)$ は各領域の節点を丁度一つずつ含むので、 G' は直NA連結である。よって、 T_1 と T_2 を連結させる枝 e を任意に一つ選んで(G' の連結性より必ず存在する)付け加えれば、DNACTの解となる。(1)終り)

(2) DNACTの解から2T2NA'の解を作る方法。

DNACTの解である木 T から適切な枝を除去して二つの木にすることで、2T2NA'の解になることを示す。使用する記号を次に定義しておく。

【定義】 2節点 x, y を結ぶ T 上の路(唯一存在)を $p(x, y)$ と表す。 T から $E(p(x, y))$ を除去したグラフにおいて、 x を含む連結部分木を $T(x, y)$ 、 y を含む連結部分木を $T(y, x)$ と表現する。 V_i に含まれる2節点を x_i^1, x_i^2 と記す。□

まず、ある枝 $e(x, y) \in E(T)$ が存在して、 $|V(T(x, y))| = |V(T(y, x))| = |V|/2$ となることを示す。この様な枝が存在しないと仮定すると、次数3以上の節点 $x \in V(T)$ が存在して、任意の $e(x, y) \in E(T)$ に対して $|V(T(y, x))| < |V|/2$ という性質を持つ。一般性を失うことなく $x = x_i^1$ とする。 $x_i^2 \in V(T(y, x_i^1))$ である様な、 x_i^1 の T における隣接点を y とおく(図2.8参照)。この時、ある $V_j \in X (j \neq i)$ に対し $V_j \subseteq V(T(x, y))$ と仮定すると、 $p(x_i^2, x_j^1), p(x_i^2, x_j^2)$ 共に x_i^1 を経由するので T の直NA連結性に反する。よってどの $V_j \in X$ に対しても $V_j \cap V(T(y, x)) \neq \emptyset$ である。すると、 $|V(T(y, x))| \geq |V|/2$ となり、仮定に反する。よって、ある枝 $e(x, y) \in E(T)$ が存在して、 $|V(T(x, y))| = |V(T(y, x))| = |V|/2$ となることが示された。

この枝(唯一存在)を $e(x, y)$ とおき、さらに一般性を失うことなく $x = x_i^1$ とする。 $x_i^2 \in V(T(x, y))$ と仮定すると、 $|V(T(x, y))| = |V(T(y, x))|$ より $V_j \subseteq V(T(y, x))$ がある $j \neq i$ に対し成立し、 $p(x_i^2, x_j^1), p(x_i^2, x_j^2)$ 共に x_i^1 を経由するので T の直NA連結性に反する。よって $x_i^2 \in V(T(y, x))$ 。次に $V_j \subseteq V(T(x, y))$ なる $j \neq i$ が存在すると仮定すると、 $p(x_i^2, x_j^1), p(x_i^2, x_j^2)$ 共に x_i^1 を経由するのでやはり T の直NA連結性に反する。よって全ての $V_j \in X$ に対し $V_j \cap V(T(y, x)) \neq \emptyset$ である。 y につ

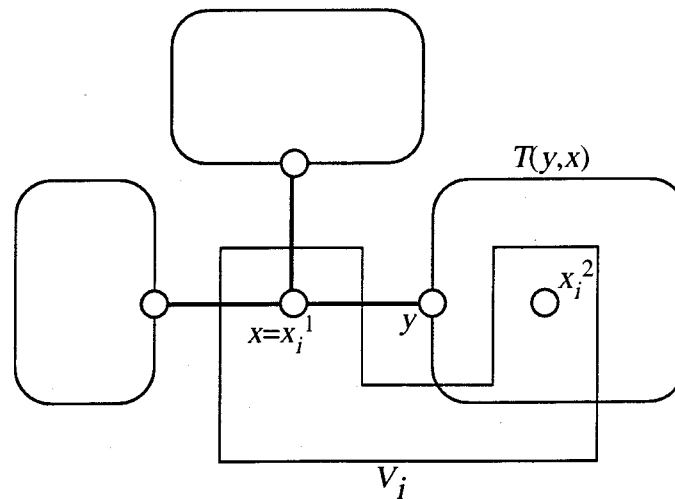


図2.8 定理2.14の証明の説明

Fig. 2.8. Proof of theorem 2.14.

いても同様な検証を行なうことによって、全ての $V_j \in X$ に対し $V_j \cap V(T(x,y)) \neq \phi$ が言える。ここで補題2.1より、 $T-e(x,y)=T(x,y) \cup T(y,x)$ はNA連結である。(2)終り) (定理2.14の証明了)

以上で直NA連結性を持つ全域部分グラフを求める問題は、枝数最小のものを求めるのは勿論、単なる直NA連結全域木を求める問題ですらNP完全であることが分かった。この結果を利用して、MDNACで $h=|V|-1$ に限定した問題もNP完全であることも証明できる。

【系2.3】 $h=|V|-1$ に限定したMDNACはNP完全である。 \square

証明) 対象を2節点領域より成り、連結な領域グラフに限り、この領域グラフ $G^+=(G=(V,E),X)$ について $h=|V|-1$ に限定したMDNACの解の存在と、DNaCTと解の存在が一致することを示す。

$h=|V|-1$ に限定したMDNACの解が存在する場合、解は全域木か複数の連結成分に分かれているかどちらかである。もし全域木が得られているならば、そのままDNaCTの解となる。複数の連結成分に分かれている場合、補題

2.1より連結成分数は2であり、それぞれの連結成分は各領域の節点を丁度一つずつ含むので、各連結成分の全域木 T_1, T_2 をとって、 T_1, T_2 間に適当な枝を一つ加えればDNACTの解となる。よってDNACTの解も存在することになる。

また、DNACTの解が存在する場合は、明かにそれが $h=|V|-1$ に限定したMDNACの解でもある。 Q.E.D.

よって次節では、 $h|V|$ 程度の枝数 (h は適当な定数) の直NA連結部分グラフを高速に得ることを考える。

2. 5. 5. アルゴリズムMODTREE

2. 2で示した様に、 k -NA連結性を満たす最小の枝数の全域部分グラフを求める問題はNP完全だが、疎なグラフ (枝数 $O(k|V|)$) で容易に実現することができる。これと同様に直NA連結性に対しても、疎なグラフで実現できると有り難い。だが元のグラフの形状によっては、疎なグラフで直NA連結性を実現できない場合がある。図2.9のグラフは完全2部グラフであり、 $|V|^2/4$ 本の枝を持つ密なグラフだが、図の領域 V_1 を考えると、どの枝を取り

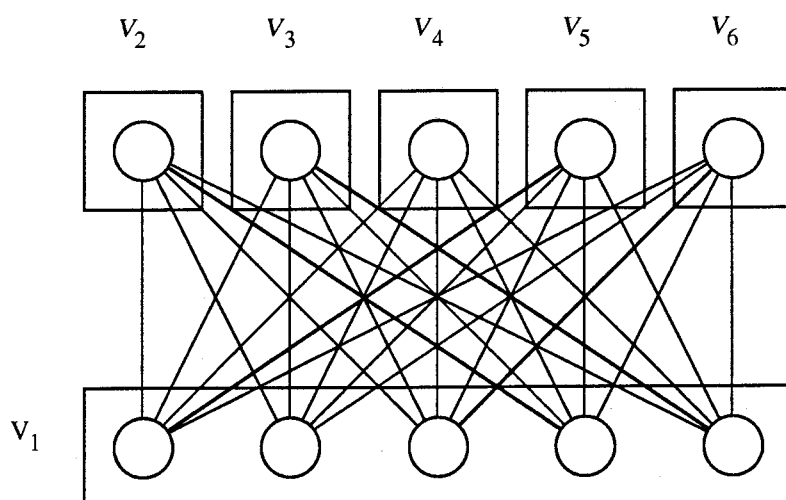


図2.9 密な極小直NA連結グラフ

Fig. 2.9. A dense minimum directly NA-connected graph.

去っても直NA連結性は失われる、すなわち直NA連結性を持つ極小領域グラフである。しかし実はこれは特殊な例であり、大部分のグラフは疎な直NA連結全域部分グラフを持つと考えられる。以下で、そのための十分条件を示すと共に、疎な直NA連結全域部分グラフを見出すアルゴリズムを示す。なお、非連結なグラフは各連結成分毎に独立に論ずれば良いので、以下で扱うグラフは全て連結とする。すなわち $|V|+|E|=O(|E|)$ を仮定する。実際、直NA連結な領域グラフ $G^{*a}=(G=(V,E),X)$ においてグラフ G は、 $|X|=1$ の場合を除いて、孤立点($E(x,V-x)=0$ の節点)を持たないので、 $|E|\geq \lceil |V|/2 \rceil$ となり、たとえ連結でなくとも、少なくとも $|V|+|E|=O(|E|)$ は成立する。

【定義】 領域グラフ $G^{*a}=(G=(V,E),X)$ において、領域 $V_i \in X$ に対し $G-V_i$ が連結でない場合、 V_i をカット領域と呼ぶ。(図2.9では領域 V_1 がカット領域である。) □

【定理2.15】 直NA連結領域グラフ $G^{*a}=(G=(V,E),X)$ が、カット領域を持たないならば、直NA連結全域部分グラフ $G'=(V,E')$ で、 $|E'| \leq 2|V|-3$ のものが存在する。□

証明) 証明は構成的に行なう。定理2.15の条件を満たす領域グラフ G^{*a} が与えられた時、枝数 $2|V|-3$ 以下の直NA連結全域部分グラフ G' を作るアルゴリズムMODTREEを以下に示す。まず領域数 $|X|$ が1の場合は、直NA連結性の定義から任意の全域部分グラフが直NA連結になるので、 (V, ϕ) を解として出力すれば良い。よって以下では $|X| \geq 2$ と仮定する。

```

procedure MODTREE
comment  $G^{*a}=(G=(V,E),X)$ : 入力 of 直NA連結領域グラフ
       $G'=(V,E')$ : 出力される全域部分グラフ;
begin
1   $E':=\phi$ 
2  任意の全域木  $T$  を一つ構成する。  $E':=E(T)$ ,  $G':=(V,E')$ 
3   $Y:=X$ 
4  while  $Y \neq \phi$  do
      begin
5       $Y$  に属する領域  $V_i$  を選ぶ。
6      if  $G'':=G'-V_i$  が非連結 then
          begin
7               $E(G-V_i)-E(G'')$  から必要最小本数の枝を選んで、  $G''$  に付け加えて、  $G''$  を連結にする。
8               $E':=E' \cup E(G'')$ 
          end;
9      if  $x \in V_i$  かつ、  $G'=(V,E')$  において  $x$  に隣接する  $V-V_i$  の節点が存在しない様な  $x$  が存在する then
          begin
10             その様な  $x$  それぞれに対して、
                   $\alpha(x,y) \in E$ ,  $y \in V-V_i$  である枝を一つ  $G'$  に加える。
          end;
11       $Y:=Y-\{V_i\}$ 
      end;
end;

```

手続きMODTREEの出力である G' が枝数 $2|V|-3$ 以下の直NA連結全域部分グラフであることを以下で示す。

まず、この手続きの各行が常に実行可能であることを示す。第2行の T を作ることは、 G が連結であるので容易である。第6行で $G'':=G'-V_i$ が非連結の場合、 G はカット領域を持たないという前提なので、第7行目の操作は実行できる。第9行で $x \in V_i$ かつ G' において x に隣接する $V-V_i$ の節点が存在しない様な x が存在する場合、第10行において、この条件を満たす $\alpha(x,y)$ は

補題2.7より必ず存在する。以上で、上の手続きが常に動作することが示された。

次に最終的に得られた G' が直NA連結性を満たすことを示す。領域 V_i に着目する。第7行の操作により、 $G'-V_i$ は連結であり、任意の $y, z \in V-V_i$ 間に、 $G'-V_i$ 上の路が存在する。第10行の操作により、各 $x \in V_i$ から、ある $y \in V-V_i$ に対して、枝 $a(x, y) \in E'$ が存在するので、どの $z \in V-V_i$ をとっても、 x, z 間に V_i-x の節点を經由しない、 G' 上の路が存在する。よって直NA連結である。

次に、枝数が $2|V|-3$ 以下であることを示す。 T の節点数を n 、枝数を $m'=n-1$ 、領域 V_i の節点数を n_i とする。また、 T の枝のうち、両端の節点共に V_i に含まれている枝の数を m_{2i} 、片方の節点のみが V_i に含まれている枝の数を m_{1i} とする。定義より、 $m'=1/2(\sum_i m_{1i}) + \sum_i m_{2i}$, $n=\sum_i n_i$ である。

T の節点 V_i で誘導される部分グラフを T_i とする。 T_i の連結成分の数は n_i-m_{2i} なので、 $T-V_i$ の連結成分の数は $m_{1i}-(n_i-m_{2i})+1=m_{1i}+m_{2i}-n_i+1$ 。よって、第7行において、 G'' の連結成分の数は $m_{1i}+m_{2i}-n_i+1$ 以下である。故に第7行で付加される枝の数は $m_{1i}+m_{2i}-n_i$ 以下。第9行において、隣接する $V-V_i$ の節点が存在しない $x \in V_i$ の数は m_{2i} 以下。よって第10行で付加される枝の数は m_{2i} 以下。故に第5行で選ばれた V_i に関して、doループ内で付加される枝の数は $m_{1i}+m_{2i}-n_i+m_{2i}=m_{1i}+2m_{2i}-n_i$ 以下。よって

$$E' \leq m' + \sum_i (m_{1i} + 2m_{2i} - n_i) = m' + 2m' - n = 2n - 3$$

となる。以上より、枝数が $2|V|-3$ 以下である。

(定理2.15の証明了)

MODTREEで得られた G' の枝数の見積もり $2|V|-3$ は厳密な上限である。つまり、枝数 $2|V|-3$ になる例がある。図2.10の例(節点数12)において、始めに T を実線の様を選び、 V_1, V_2, V_3 の順に操作すると、全ての枝が選ばれ枝数は $21=2 \times 12 - 3$ となる。但し最適解は太線の枝で、枝数は8である。

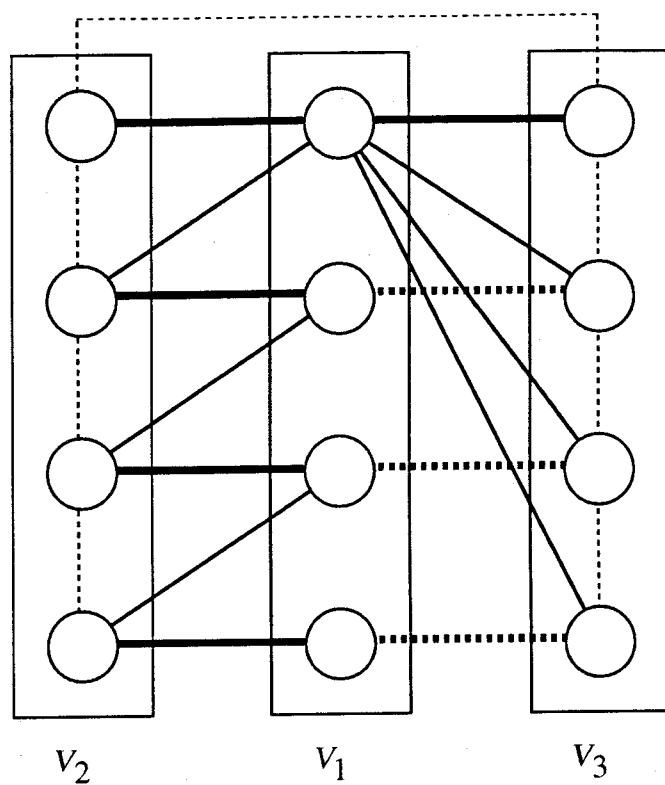


図2.10 最悪枝数 $|E'|=2|V|-3$ を与える問題例

Fig. 2.10. A worst case instance of $|E'|=2|V|-3$.

2. 5. 6. 近似誤差

次に手続きMODTREEの近似誤差を見積る。2. 5. 4で述べた様に、MNACの最適解の下限 $|V| - \min(|V_i|)$ は、MDNACの最適解の下限でもあるので、本アルゴリズムの相対近似誤差は

$$\frac{2|V| - 3}{|V| - \min(|V_i|)} \leq \frac{2|V| - 3}{|V| - |V|/|X|} \leq \frac{|X|(2-3/|V|)}{|X|-1} < \frac{2|X|}{|X|-1}$$

となる。この式は、 $|X|$ が小さいほど、押さえられる誤差が大きくなることを意味する。しかし $|X|=1, 2$ の場合には、以下に示す様に、簡単に最適解もしくは良い近似解が得られる。

$|X|=1$ の場合には、定理2.15の証明でも述べた様に、最適解 $G'=(V, \phi)$ が自明に得られる。

$|X|=2$ の場合には、 $X=\{V_1, V_2\}$ とすると、領域グラフ $G^{*a}=(G=(V,E),X)$ が直NA連結であるならば、補題2.7より、任意の節点 $x \in V$ に対し、 $E(V_1, V_2)$ に属する枝のうち、少なくとも一本は x に接続している。よって、全域部分グラフ $(V, E(V_1, V_2))$ の極大全域森 (V, F) を考えると、任意の節点 $x \in V$ と x を含まない唯一の領域 V_i 間の枝が必ず F に存在する。よって (V, F) は直NA連結である。 (V, F) を構成することは、線形時間で可能である。また、 $|F| \leq |V|-1$ より相対近似誤差は

$$\frac{|V| - 1}{|V| - \min(|V_i|)} \leq \frac{|V| - 1}{|V| - |V|/2} \leq 2(1-1/|V|) < 2$$

で押さえられる。MODTREEの前処理に、 $|X|=1$ か $|X|=2$ か $|X| \geq 3$ かの判定を行なって、 $|X|=1$ か $|X|=2$ ならば上記の処理を行なう様にすれば、最適解もしくは相対近似誤差2以下の解を得る。

以上から、 $|X|=3$ の場合に相対近似誤差が3となり最大である。実際、図2.10の例で各領域内の節点数を増加させると、誤差が3に漸近する。領域内の節点数が定数の場合には、 $|V|$ の増加に従って $|X|$ も増加するので、相対近似誤差は2に漸近する。

2. 5. 7. 計算量

次にMODTREEの計算量を見積る。

まず、各節点に接続する枝のデータを、リストで作成しておく。こうするとアルゴリズム2行目の全域木 T の作成は $O(|E|)$ で出来る。4行目の繰り返しは領域数($|X| \leq |V|$)回行なわれる。6行目の G' の連結性の判定は $O(|E|)$ で行なえ、その際に、 G'' において同一連結成分に属する節点を縮約しておく(リストのポインタの付け替えですむので、 $O(|E|)$ で可能)。すると、第7行目はその縮約したグラフにおける全域木を作成することに等しいので、 $O(|E|)$ で可能である。第9行目の該当節点の判定はアルゴリズムを通して $O(|V|)$ であり、第10行目の枝の付加はアルゴリズムを通して $O(|E|)$ である。

以上から、本アルゴリズムの計算量は $O(|E||X|)$ ($\leq O(|V||E|)$) である。

2. 5. 8. カット領域の有る場合

以上で述べたアルゴリズムはカット領域が無いという前提で述べた。通信網等応用を考えた場合、この条件はかなりの場合に成立すると考えられるが、アルゴリズムとしてはカット領域が存在しても正確に作動することが望ましい。そこで、カット領域が存在する場合に対応できる様にMODTREEを改良する。

カット領域が有る場合、MODTREEの第8行目の処理において、 G'' を連結に出来ない場合がある。しかし、補題2.6より、領域 $V_i \in X$ と節点 $x \in V_j$ ($i \neq j$) が直NA連結である必要十分条件は、 $G - (V_j - x)$ における、 x を含む連結成分が、 V_i 内の節点を少なくとも一つ含むことである。よって、 $G - V_j$ において、 x との間に枝がある(つまり $|E(x, V(G_j^k); G)| \geq 1$ である)様な連結成分を G_j^k , $k=1, 2, \dots, K_j$ とすると、各 G_j^k について x と G_j^k の間の枝を、少なくとも1本は G'' に加える様にすれば、得られるグラフは直NA連結になる。このことを考慮して、6～8行目を以下の様に変更できる。

```

6      if  $G'' := G' - V_i$  が非連結 then
      begin
7           $E(G - V_i) - E(G'')$  から必要最小本数の枝を選んで  $G''$  に付け加えて、
           $G''$  の連結成分数を最小にする。
8-1       $E' := E' \cup E(G'')$ 
          end;
8-2      if  $G'' := G' - V_i$  が非連結 then
          begin
8-3          for each  $x \in V_i$  do
8-4               $x$  に隣接する節点  $y \in V - V_i$  を一つでも含む連結成分  $G_i^k$ ,  $k=1, 2, \dots, K_i$  を作成し、全ての  $G_i^k$ ,  $k=1, 2, \dots, K_i$  に対して、 $\exists y \in V(G_i^k), e(x, y) \in E$  を一つずつ選び、 $E' := E' \cup e(x, y)$  とする。;
          end;

```

この改良により、カット領域がある場合にも直NA連結な全域部分グラフを得ることができる。

本改良版の計算量であるが、第7行は、改良前のMODTREEの第7行で行なったのと同様に、 G'' の連結成分を縮約し、その縮約グラフに枝集合 $E(G - V_i) - E(G'')$ を付加したグラフ上に対し、極大全域森を作成することで達成される。よってこの部分の計算量は改良前と同じである。第8-2行の判定は、第7行を実行した際に同時に行なえる。第8-4行の操作は、手続き DECIDE-Dの第8、9行と同様に、連結成分毎の節点のリストと、各点の含まれる連結成分を示す配列を準備する（一つの V_i につき $O(|E|)$ ）ことで、アルゴリズム全体で $O(|E|)$ でできる。よって、改良版のMODTREEの計算量は改良前と同じく、 $O(|E||X|)$ ($\leq O(|V||E|)$) である。

2. 6. まとめ

本章では、通信網の形態をより現実に合わせてモデル化するために、領域グラフという概念を導入し、節点と領域間の連結度であるNA連結度を定義した。NA連結度の概念を用いることにより、通信網の強度をより正確に測ることができる。そして本章では、NA連結度に関するいくつかの問題がNP完全になること、しかしそれらの問題に絶対近似誤差あるいは相対近似誤差の上限を評価できる高速 ($O(|V|+|E|)$ 時間) 近似アルゴリズムが存在することを示した。そのために「 k -連結性を保存する部分グラフは、 k -NA連結性も同時に保存する」という定理を証明した。連結性を保存する全域部分グラフについては2連結に限れば、極小なものを求めるアルゴリズム [HAN 92] や相対近似誤差が $5/3$ あるいは $3/2$ のものを求めるアルゴリズム [KHUL 92] [GARG 93] が提案されている。これらも2-NA連結に適用できることが上記の定理から予想される。

また、上記の定理は領域間の連結度にも拡張できる。さらに、NA連結は節点連結の拡張であったが、点連結と辺連結を含む概念である混合カットに基づいてNA連結の一般化を行なうことも考えられ、これらにも同様の法則が成立することも示した。

次に大規模事故に対応できる強度の概念として直NA連結を定義し、任意の直NA連結グラフに対して、「カット領域が無い」という制約の下では、枝数 $2|V|-3$ 以下の直NA連結全域部分グラフが存在し、そのうちの一つを計算時間 $O(|E||X|) \leq O(|V||E|)$ で求めるアルゴリズムを示した (領域数が定数ならば、線形時間 $O(|E|)$ である)。カット領域とは、グラフを分断する領域のことで、カット領域が無いという条件は実網をモデル化したグラフでは、自然な条件である。

さらに理論上の完全を記すため、上のアルゴリズムをカット領域のあるグラフにも適用できる様に改良を加えた。この場合に必要な枝数の上限は巧く押さえられないが、図2.9の様にカット領域のあるグラフでは、枝数 $O(|V|^2)$ を持つ極小な直NA連結グラフも存在しうることから止むを得ない。最小枝数の直NA連結全域部分グラフを求める問題は、NP完全であるうえ、枝数 $|V|-1$ 以下で実現できるかどうかを判定する問題もNP完全である。よって、本稿で示した、任意のカット領域の無い直NA連結領域グラフに対して、枝数 $2|V|-3$ 以下の直NA連結な部分グラフが存在し、その一つを $O(|E||X|)$ で求

めることができるという結果は、重要である。

次に今後の研究課題について述べる。

与えられた領域グラフが k -NA連結であるか否かの判定のアルゴリズムを、グラフが k -連結であるか否かの判定と同程度の計算量にまで高速化できる可能性がある。また、NA連結が定義されたことによって、連結性に関する諸問題がNA連結に拡張される。例えば、NAカットの様な概念を定義し、最小NAカットを計算する高速アルゴリズムの存在が予想される。通信網では、この他に経路長の制限や、経路選択方式に依存する制約が加わることがあり[INOUE 92]、それらを考慮した連結度の検討[SATA 93]も今後行なっていく必要がある。特に通信網では、たとえ局間に経路が存在しても、その経路が余りにも長いと品質や安全性の面で問題があるので、連結度だけではなく直径も考慮することが重要である。よって直径と連結度を同時に考慮する全域部分グラフに関する問題は実用的にも、理論的にも重要と考えられる。例えば、与えられた領域グラフの直径とNA連結度の制約を満足しながら、枝数を少なくするという問題が考えられる。またその際に、NA連結度に対応して、NA距離という概念を定義して考慮することも意味深い。

直NA連結については、カット領域のある場合の近似誤差を押さえる方法も重要である。また、 k -直NA連結への発展の見込みもある。

また本章では、与えられたグラフから枝を削除する形式の問題のみを扱ったが、枝を付加する形式の問題も同様に重要であるので、検討の必要がある。例えば、 k -NA連結性を満たしていない領域グラフに、最小数の枝を付与して k -NA連結性を満足させるという問題がある。その際枝を付与できる場所に制限があれば、NP完全問題であるが、任意の節点間に付与できるならば、多項式時間でできる可能性がある。

本章で定義した領域グラフは、領域が節点集合の分割であると定義したが、節点集合の任意の部分集合族を認める様に拡張すると、領域は超グラフの枝と同じになる。よって、超グラフで得られた性質を領域グラフに適用できる可能性もあり、この観点からの研究の発展も興味あるところである。

第3章

制約付き多品種流問題

3. 1. 通信網と多品種流問題

2章では、通信網のモデルをグラフとして扱い、経路数で網の強度を測った。現実の通信網では、各枝は通信容量など固有の特性をもつのが普通であり、従って枝毎に太さ（容量）を表す関数を対応させて、経路の太さも考慮にいられて評価すればより詳しい評価ができる。

経路に太さを考える問題で代表的なのが、最大流問題(maximum flow problem)、最小費用流問題(minimum cost flow problem)等の流れの問題[LAWL 76] [AHUJ 89] [IWAN 93]である。最大流問題は各枝に容量（流量の上限）が与えられており、与えられた2点間に最大量の流れを運ぶ問題である。最小費用流問題は各枝に流量を引数とする費用関数が与えられている場合、指定された2点間に、与えられた流量を費用最小で運ぶ問題である。費用関数が線形であれば、最小費用流問題は線形計画問題(LP)に定式化できるのでクラスPに属する[KHAC 79] [KARM 84]。現在ではさらに強多項式時間^{*1}のアルゴリズムまで発見されている[TARD 85]。

上記の流れの問題では通常流れに区別を付けず、一種類の流れしか無いとしているが、通信網や交通流等への応用を考えると、違う目的地着の通信を同じに扱うことは出来ず、それらを表す流れを区別して扱わざるを得ない。この様に流れに複数の種類がある問題を多品種流問題(MCF: multi-commodity flow problem)と呼ぶ。多品種流問題に関しても色々研究がなされている[LAWL 76] [KENN 80] [FUKU 84] [NAGA 88] [NAGA 89] [AWER 93a] [AWER 93b]。多品種流問題にも、各枝の容量を満たす範囲で各品種の要求量を流すことができるかを判定する問題（本稿では特に断わらない限りMCFはこの問題を意味する。）や、費用最小の流し方を求める

^{*1} ネットワークを表すグラフ $G=(V,E)$ の節点数 $|V|$ と枝数 $|E|$ のみの多項式で表されるオーダーの計算時間を持つアルゴリズムを強多項式時間(strongly polynomial time)アルゴリズムと言う。

問題である最小費用多品種流問題(CMCF: minimum cost multi-commodity flow problem)があるが、これらも(費用関数が線形ならば)LPに定式化できるためクラスPに属することが知られている。しかしLPに定式化して解いたのでは計算時間が多く実用的でない。そこで、流れの問題の特性を利用した高速解法が必要となってくる。最近、品種数の多い多品種流問題の効率的なアルゴリズム[NAGA 89]や、枝数のプロセッサで並列化出来る ϵ -近似^{※1}高速アルゴリズム[AWER 93a]等が提案され、多品種流問題によって解くことができる問題の範囲が広がった。

しかし通信網に関する現実の事例を多品種流問題に定式化した場合、信頼性や制御の容易性などの理由から、さらに種々の制約条件が付加されることが多い。例えば、管理の煩雑さを避けるために通信経路の候補を絞り込む、つまり(1)一つの品種の流れが通る路の総数に上限を設ける[ASH 90] [ITO 92c]という制約、また、通信経路が長くなることによる品質劣化を防ぐ[GIMP 74] [ASH 92] [INOUE 92] [ITO 92c]、つまり(2)各品種の流れる路の経由する枝数に上限を設けるという制約、また、通信が交換機の自律的な制御で通信経路が決定されるため、(3)各品種の経由する路の集合は指定できても、その中を流れる量は、ある関数で自然に決まってしまうという制約[INOUE 92]、等が考えられる。これらの制約を加えると、従来のアルゴリズムが適用できなくなってしまう。実際、(1)は[ELSA 88]によって、経路数制限が1の場合にNP完全になることが証明された。任意の経路数制限数についても、その結果は容易に拡張できる。

本章では(2)と(3)の制約についての検討結果を記述する。すなわち、(2)については3.2で、任意の経由枝数制限値に対する多項式時間アルゴリズム[ITO 92b]を示す。(3)については、流量配分則が均等配分の場合[ITO 94d]について、経由枝数制限値も考慮して3.3で扱う。

^{※1} 問題例が実行可能解を持つとき、任意の正実数 $\epsilon > 0$ に対して、各品種 k の要求量 $d(k)$ を $(1+\epsilon)^{-1}$ 倍した問題例の実行可能解を求めることができるアルゴリズムを ϵ -近似アルゴリズムという。 ϵ を小さくすることで、理論的にはいくらでも最適解に近い解を見つけることができる。その際の計算時間が ϵ^{-1} を含む多項式で表される時、その ϵ -近似アルゴリズムは多項式時間アルゴリズムである、と考える。

3. 2. 経路枝数制限の付いた問題

多品種流問題(MCF)を現実の通信網に適用する場合、信頼性や制御の容易性などの理由から、各流れが経由する路の枝数に上限を付ける必要がある場合が多い[GIMP 74] [ASH 92] [INOUE 92] [ITO92c]。以下この問題を経路枝数制限付き多品種流問題(MCFL: multi-commodity flow problem with link number constraint)と呼ぶ事にする。MCFLは、経路枝数の上限値 l が小さい場合でしかも定数に固定してある場合(例えば $l=2$)には適用可能なアルゴリズムが検討されている[ONO 91]が、一般の l についての検討はなされていない。実際、小さい l に関しては、可能な全ての路を変数として列挙してLPに定式化することも可能であるが、この方法だと、 l の増加に従って変数量が指数関数的に増大する。

本報告では、まずMCFLが(l を変数としても)クラスPに属することを、MCFLを、線形費用の最小費用多品種流問題(CMCF)に帰着する事により示す(CMCFがクラスPに属することは、それをLPに帰着できることから分かる)。また、この帰着手順に基づくアルゴリズムを提案する。本アルゴリズムを用いれば、CMCFの効率的なアルゴリズムを、MCFLに応用することが可能である。

3. 2. 1. 枝数制約付き多品種流問題の定義

【定義】

$N=(G,K,S,d,u,c)$: ネットワーク

$G=(V,A)$: 有向グラフ

K : 品種(commodity)集合

$S=\{(s(k),t(k))|k \in K\}$: 全品種の流出点(source)、流入点(sink)の集合

$s(k),t(k) \in V$: 品種 k の流出点、流入点。ただし、全ての $k \in K$ について $s(k) \neq t(k)$ 。

$d(k) \in \{0 \text{ 以上の実数}\}$: 品種 $k \in K$ の需要量(demand)

$u(a) \in \{0 \text{ 以上の実数}\}$: 枝 $a \in A$ の容量(capacity)

$c(a) \in \{0 \text{ 以上の実数}\}$: 枝 $a \in A$ の費用係数(cost coefficient)

$f(a,k) \in \{0 \text{ 以上の実数}\}$: 流れ(flow)。枝 $a \in A$ 上の品種 $k \in K$ の流れの値を

表す。

流量保存則：任意の $x \in V$ と任意の $k \in K$ に対し

$$\sum_{a(x,y) \in A} f(a(x,y),k) - \sum_{a(y,x) \in A} f(a(y,x),k) = \begin{cases} d(k) & (x=s(k)) \\ -d(k) & (x=t(k)) \\ 0 & (x \neq s(k), t(k)) \end{cases}$$

容量制約：任意の $a \in A$ に対し

$$\sum_{k \in K} f(a,k) \leq u(a)$$

流量保存則と容量制約の両方を満たす流れ f を実行可能流と呼ぶ。□

【定義】 最小費用多品種流問題(CMCF)

[入力] ネットワーク $N=(G=(V,A),K,S,d,u,c)$ 。

[要請] 実行可能流 $f(a,k)$, $a \in A$, $k \in K$ のうち、総費用 $\sum_{a \in A} c(a)F(a)$ の最小なものを求めよ。但し、 $F(a)=\sum_{k \in K} f(a,k)$ は枝 a 上の全品種の流れの和を表す。□

CMCFはLPに定式化できるため、クラスPに属する。

【定義】 ネットワーク $N=(G=(V,A),K,S,d,u,c)$ 上の流れ（実行可能流とは限らない） $f(a,k)$, $a \in A$, $k \in K$ に対し、品種 $k_0 \in K$ と、 $s(k_0)$, $t(k_0)$ 間の路 $p=\langle a_1, a_2, \dots, a_\ell \rangle$, $a_1, a_2, \dots, a_\ell \in A$ とが存在して、以下の2条件を満たすとする。

(1) ある実数 $0 \leq v \leq d(k_0)$ が存在して、 $f(a_1, k_0)=f(a_2, k_0)=\dots=f(a_\ell, k_0)=v$ である。

(2) $k \in K - k_0$ または $a \in A - A(p)$ ならば、 $f(a,k)=0$ である。

このとき、流れ f を単純流(simple flow)と呼び、 p を単純流 f の通る路と呼ぶ。また、 p の長さ ℓ を単純流 f の経由枝数と呼ぶ。(1)の v の値を単純流 f の流量といい、 k_0 を f の品種という。

流れ $f(a,k)$ が、経由枝数 ℓ 以下の単純流の和によって表すことができること、すなわち、経由枝数 ℓ 以下の単純流 $f_j(a,k)$, $j=1, 2, \dots, J$ が存在して、全ての枝 $a \in A$ と全ての品種 $k \in K$ について

$$f(a,k) = \sum_{j=1}^J f_j(a,k)$$

を満たすとき、流れ f の経由枝数は ℓ 以下であると言う。また、 f は単純流 $f(a, k)$, $j=1, 2, \dots, J$ に分解(resolute)できるという。□

例えば図3.1(a)の流れ f は(b), (c), (d), (e)に示す 4 つの単純流 f_1, f_2, f_3, f_4 の和で表すことができる。これらの単純流の経由枝数は各々 2, 3, 3, 2 であるので、 f の経由枝数は 3 以下である。

最小費用多品種流問題CMCFと単純流の間に次の関係がある。

【補題3.1】 CMCFに最適解（最小費用流）が存在するならば、単純流に分解できる最適解が存在する。□

本補題の証明は付録 1 に記す。本節で考える問題は次の様に定義される。

【定義】 経由枝数制限付き多品種流問題(MCFL)

[入力] ネットワーク $N=(G, K, S, d, u, 0)$, $\ell \in \{\text{自然数}\}$ 。但し 0 は全要素が 0 のベクトルを表す。

[要請] 経由枝数 ℓ 以下の実行可能流 $f(a, k)$, $a \in A$, $k \in K$ を求めよ。□

MCFLでは費用係数 c を入力しないので、0ベクトルで表現した。

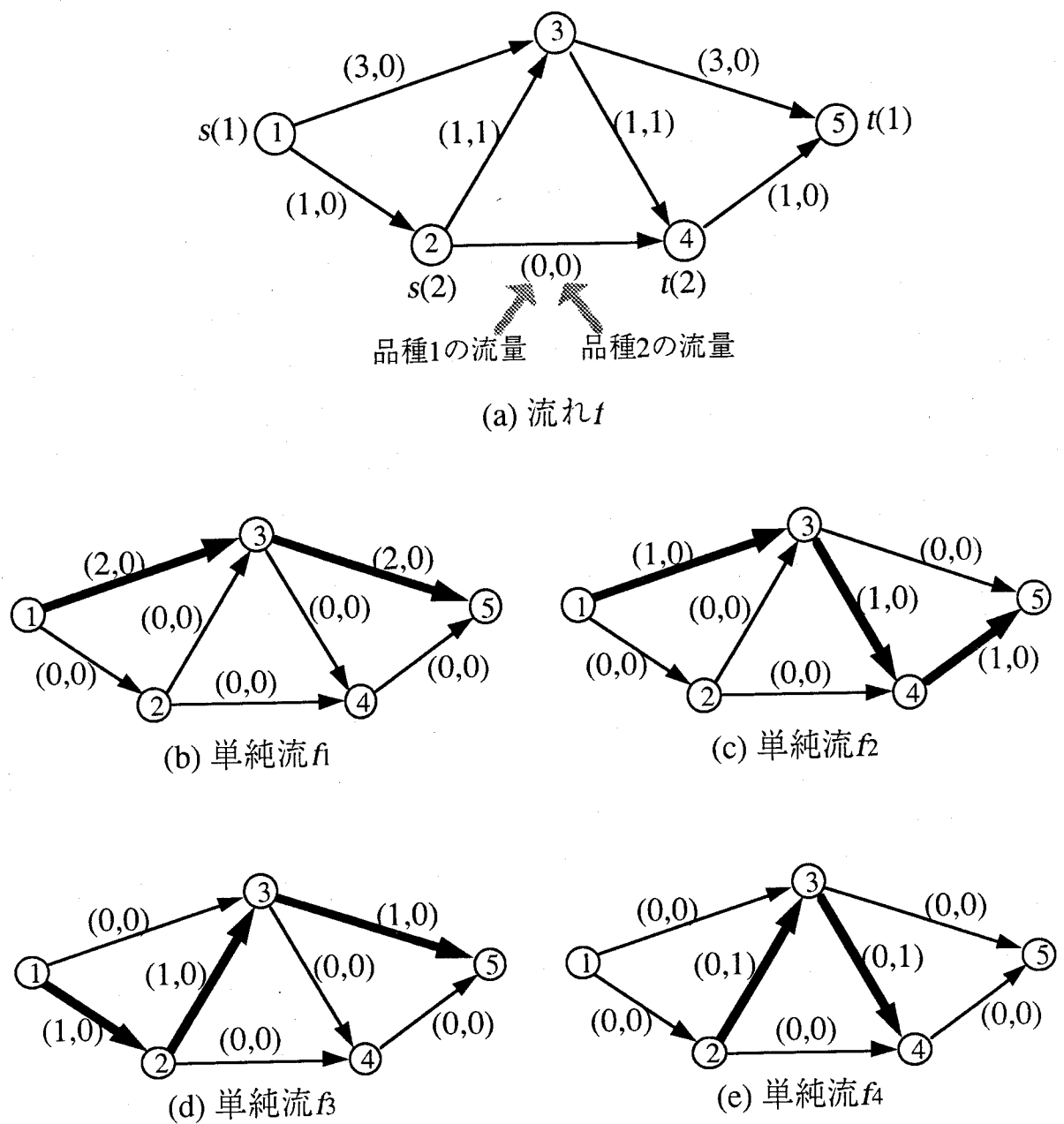


図3.1 流れ f の単純流分解

Fig. 3.1 A resolution of flow f into simple flows.

3. 2. 2. 円環化法

本節では次の操作によって、MCFLがCMCFに帰着されることを示す。図3.2にこの操作の例を記しておく。

【定義】 $N=(G=(V,A),K,S,d,u,0)$ に対し次のネットワーク $N^l=(G^l,K^l,S^l,d^l,u^l,c^l)$ を N の l 段円環ネットワーク (torus network) と呼ぶ。但し l は自然数である。

$$G^l=(V^l,A^l)$$

$$V^l=V_0 \cup V_1 \cup \cdots \cup V_{l-1}$$

$$V_i=\{x_i \mid x \in V\}, i \in \{0, 1, \cdots, l-1\}$$

$$A^l=A_0 \cup A_1 \cup \cdots \cup A_{l-1}$$

$$A_i=A_i^1 \cup A_i^2, i \in \{0, 1, \cdots, l-1\}$$

$$A_i^1=\{a(x_i, y_{(i+1) \bmod l}) \mid a(x, y) \in A\}, i \in \{0, 1, \cdots, l-1\}$$

$$A_i^2=\{a(x_i, x_{(i+1) \bmod l}) \mid x \in V\}, i \in \{0, 1, \cdots, l-1\}$$

$$K^l=\{k_i \mid k \in K, i \in \{0, 1, \cdots, l-1\}\}$$

$$S^l=\{(s^l(k_i), t^l(k_i)) \mid k \in K, i \in \{0, 1, \cdots, l-1\}\}$$

$$s^l(k_i)=s(k)_i, k \in K, i \in \{0, 1, \cdots, l-1\}$$

$$t^l(k_i)=t(k)_i, k \in K, i \in \{0, 1, \cdots, l-1\}$$

$$d^l(k_i)=d(k), k \in K, i \in \{0, 1, \cdots, l-1\}$$

$$u^l(a(x_i, y_{(i+1) \bmod l}))=u(a(x, y)), a(x_i, y_{(i+1) \bmod l}) \in A_i^1, i \in \{0, 1, \cdots, l-1\}$$

$$u^l(a(x_i, x_{(i+1) \bmod l}))=\infty, a(x_i, x_{(i+1) \bmod l}) \in A_i^2, i \in \{0, 1, \cdots, l-1\}$$

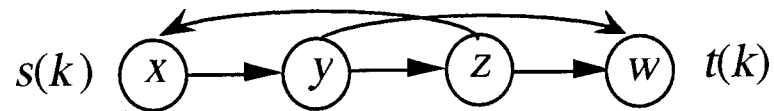
$$c^l(a)=1, a \in A_0$$

$$c^l(a)=0, a \in A^l - A_0$$

ネットワーク N に基づいて円環ネットワーク N^l を作ることを円環化 (torusize) と呼ぶ。□

円環ネットワークの直感的な説明をしておく。 $G^l=(V^l, A^l)$ は次の様にして $G=(V, A)$ から作られる。 $G=(V, A)$ の複製 l 個 $G_i=(V_i, A_i), i=0, 1, \cdots, l-1$ を作って G_i の上に G_{i+1} が来るように重ね、 G_i は i 階にあると考える。但し、 $l-1$ 階の上には 0 階がつながって、輪になっているものとする。そして、各枝の頭を、真上の階の対応する節点に来るようにずらし、それから、下の階から上の階に、対応する節点を結ぶ枝をつける。品種も階に対応して l 個複製する。

Network : N



Torus Network : N^3

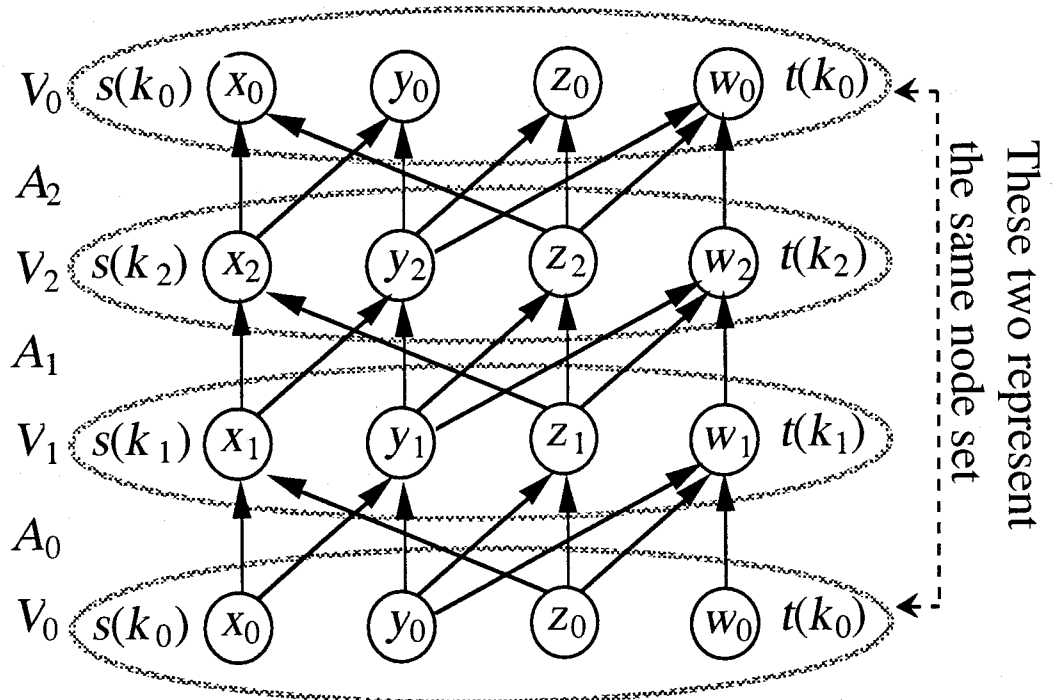


図3.2 ネットワークの円環化の例

Fig. 3.2. An example of torusized network.

こうして得られた円環ネットワーク上を運ばれる流れは、一つ枝を通過する度に、階を一つ上がることになる。各品種の流出点と流入点は同じ階にある、ということは ℓ の倍数階離れていることになる。よって、流れの経由枝数は ℓ の倍数になる。これを丁度経由枝数 ℓ の流れのみにするために、費用関数を導入し、経由枝数が長くなると、費用が大きくなる様にしたのである。次の定理でこのことを厳密に証明する。

【定理3.1】 ネットワーク $N=(G=(V,A),K,S,d,u,0)$ 、 $\ell \in \{\text{自然数}\}$ 、 $D=\sum_{k \in K} d(k)$ 、 N を円環化したネットワークを $N'=(G'=(V',A'),K',S',d',u',c')$ とすると、MCFLの問題例 (N,ℓ) が解を持つことの必要十分条件は、CMCFの問題例 N' の最適解の総費用が ℓD となることである。□

本定理の証明に必要な性質を補題として示しておく。

【補題3.2】 定理3.1の N' と D に対し、CMCFの問題例 N' の最適解の総費用は ℓD 以上である。□

証明) 補題3.1より、問題例 N' の最適解 f で、単純流に分解できるものが存在する。 f の単純流分解を、 f_1, f_2, \dots, f_H とする。費用は流量の線形関数なので、各 $f_h (h=1, 2, \dots, H)$ 毎に独立に費用を計算し、その和で f の費用を表すことができる。 $h=1, 2, \dots, H$ について、 f_h の通る路を p_h 、流量を v_h 、経由枝数を ℓ_h とする。 N' の構成法より、全ての $k_i \in K'$ について、 $s(k_i), t(k_i)$ 間の路は長さが ℓ の自然数倍である。よって $\ell_h = m_h \ell$ と書ける、但し m_h は自然数である。すると p_h は A_0 の枝を丁度 m_h 個含むので、 p_h の作る費用は $m_h v_h$ である。よって f の総費用は

$$\sum_{h=1}^H m_h v_h \geq \sum_{h=1}^H v_h = \ell D$$

となる。 Q.E.D.

定理3.1の証明) まず必要性、即ち「MCFLの問題例 (N,ℓ) が解を持つならば、CMCFの問題例 N' の最適解の総費用が ℓD である」ことを示す。

f をMCFLの問題例 (N,ℓ) の解とする。 f は、経由枝数が ℓ 以下であるような単純流 f_1, f_2, \dots, f_H に分ける事ができる。この内の品種が k であるものを $f_1^k, f_2^k, \dots, f_{H(k)}^k$ とする。 f_h^k の流量を v_h^k とし、 f_h^k の通る路を $p_h^k = \langle x(k,h,0), x(k,h,1),$

$\cdots, x(k, h, \ell(k, h)) \rangle, x(k, h, 0), x(k, h, 1), \cdots, x(k, h, \ell(k, h)) \in V$ とする、但し $\ell(k, h)$ ($\leq \ell$) は p_h^k の経路枝数である。 p_h^k に基づいて $G^\ell = (V^\ell, A^\ell)$ 上の路 $q_h^{k,i}(k)$, $i=0, 1, \cdots, \ell-1$ を

$$q_h^{k,i}(k) = \langle x(k, h, 0)_{(i+1) \bmod \ell}, x(k, h, 1)_{(i+1) \bmod \ell}, \cdots, x(k, h, \ell(k, h))_{(i+\ell(k, h)) \bmod \ell},$$

$$x(k, h, \ell(k, h))_{(i+\ell(k, h)+1) \bmod \ell}, x(k, h, \ell(k, h))_{(i+\ell(k, h)+2) \bmod \ell}, \cdots, x(k, h, \ell(k, h))_{(i+\ell(k, h)+\ell-1) \bmod \ell} \rangle$$

の様に構成する。 $q_h^{k,i}$ は G^ℓ 上の路であることは、 $j=0, 1, \cdots, \ell(k, h)-1$ に対して、 $a(x(k, h, j), x(k, h, j+1)) \in A$ より、

$$a(x(k, h, j)_{(i+j) \bmod \ell}, x(k, h, j+1)_{(i+j+1) \bmod \ell}) \in A_{(i+j) \bmod \ell}^1 \subset A^\ell$$

であり、 $j=\ell(k, h), \ell(k, h)+1, \cdots, \ell-1$ に対して、

$$a(x(k, h, \ell(k, h))_{(i+j) \bmod \ell}, x(k, h, \ell(k, h))_{(i+j+1) \bmod \ell}) \in A_{(i+j) \bmod \ell}^2 \subset A^\ell$$

であることから明かである。全ての $q_h^{k,i}$ の長さは ℓ である。各 $q_h^{k,i}$ を通る単純流 $g_h^{k,i}$ を

$$g_h^{k,i}(a, k_i) = v_h^k, a \in A(q_h^{k,i}), k_i \in K^\ell$$

の様に構成する。 p_h^k の始終点 $x(k, h, 0)$ と $x(k, h, \ell(k, h))$ は単純流の定義から、 $x(k, h, 0) = s(k)$, $x(k, h, \ell(k, h)) = t(k)$ である。よって、 $q_h^{k,i}$ の始終点は各々 $x(k, h, 0)_i = s(k)_i = s^\ell(k_i)$ と $x(k, h, \ell(k, h))_i = t(k)_i = t^\ell(k_i)$ となる。しかも $g_h^{k,i}(a, k_i) = v_h^k \leq d(k) = d^\ell(k_i)$ であるので、 $g_h^{k,i}$ は単純流の条件を満たしている。次に $g_h^{k,i}$ に基づいて N^ℓ 上の流れ g を次の様に構成する。

$$g(a, k_j) = \sum_{h=1}^{H(k)} g_h^{k,i}(a, k_j), a \in A^\ell, k_j \in K^\ell$$

ここで、 g が N^ℓ の実行可能流になっている、すなわち流量保存則と容量制約を満たしていることを示す。構成法より、全ての $a(x_i, y_{(i+1) \bmod \ell}) \in A_i^1, k_j \in K^\ell$ について、

$$\sum_{j=0}^{\ell-1} g(a(x_i, y_{(i+1) \bmod \ell}), k_j) = f(a(x, y), k) \quad (3.1)$$

となる。また、 p_h^k が単純路であることから、

$$\sum_{a(s(k), y) \in A} f(a(s(k), y), k) = d(k), k \in K \quad (3.2)$$

$$\sum_{a(y, s(k)) \in A} f(a(y, s(k)), k) = 0, k \in K \quad (3.3)$$

$$g(a(s(k)_i, y_{(i+1) \bmod \ell}), k_i) = f(a(s(k), y), k), a(s(k), y) \in A, k \in K, i \in \{0, 1, \cdots, \ell-1\} \quad (3.4)$$

$$g(a(y_{(i-1) \bmod \ell}, s(k)_i), k_i) = 0, a(y, s(k)) \in A, k \in K, i \in \{0, 1, \cdots, \ell-1\} \quad (3.5)$$

$$g(a(s(k)_i, y_{(i+1) \bmod \ell}), k_j) = 0, a(s(k), y) \in A, k \in K, i \neq j, i, j \in \{0, 1, \cdots, \ell-1\} \quad (3.6)$$

$$g(a(s(k)_i, s(k)_{(i+1) \bmod \ell}), k_j) = 0, a(s(k), y) \in A, k \in K, i, j \in \{0, 1, \cdots, \ell-1\} \quad (3.7)$$

が成立する。 $g_h^{k,i}$ が単純流であるので、各品種 k_i の流量保存則は、その流出

点 $s^{\ell}(k_i)$ と流入点 $t^{\ell}(k_i)$ を除いて成立しているのは明か。品種 $k_i \in K^{\ell}$ のその流出点 $s^{\ell}(k_i)$ における流量保存則を考えると、

$$\begin{aligned} & \sum_{a(s^{\ell}(k_i), y) \in A^{\ell}} g(a(s^{\ell}(k_i), y), k_i) - \sum_{a(y, s^{\ell}(k_i)) \in A^{\ell}} g(a(y, s^{\ell}(k_i)), k_i) \\ &= \sum_{a(s(k), y) \in A} g(a(s(k)_i, y_{(i+1) \bmod \ell}), k_i) \\ & \quad - \left(\sum_{a(y, s(k)) \in A} g(a(y_{(i-1) \bmod \ell}, s(k)_i), k_i) + g(a(s(k)_{(i-1) \bmod \ell}, s(k)_i), k_i) \right) \\ &= \sum_{a(s(k), y) \in A} f(a(s(k), y), k) \quad ((3.4), (3.5), (3.7) \text{より}) \\ &= d(k) \quad ((3.2) \text{より}) \\ &= d^{\ell}(k_i) \end{aligned}$$

よって、品種 $k_i \in K^{\ell}$ のその流出点 $s^{\ell}(k_i)$ における流量保存則は満たされる。残る品種 $k_i \in K^{\ell}$ の流量保存則は流出点 $t^{\ell}(k_i)$ におけるもののみであるが、流量保存則の性質から、一節点のみ流量保存則が成立しないことはできないので、 $t^{\ell}(k_i)$ においても成立することが分かる。以上から、流量保存則の成立が確かめられた。

次に容量制約を満たすことを確認する。 $a \in A_i^2, i=0, 1, \dots, \ell-1$ に関しては、容量が無限大なので、当然容量制約は満足している。 $a(x_i, y_{(i+1) \bmod \ell}) \in A_i^1, i=0, 1, \dots, \ell-1$ における容量制約は、

$$\begin{aligned} & \sum_{k \in K} \sum_{j=0}^{\ell-1} g(a(x_i, y_{(i+1) \bmod \ell}), k_j) = \sum_{k \in K} f(a(x, y), k) \quad ((3.1) \text{より}) \\ & \leq u(a(x, y)) = u^{\ell}(a(x_i, y_{(i+1) \bmod \ell})) \end{aligned}$$

よって容量制約も満足する。以上から、 g は N^{ℓ} の実行可能流である。

次に、 g による費用を計算する。費用は流量の線形関数なので、単純流 $g_h^{k,i}$ 毎に独立に費用を計算し、その総和によって g の費用を求めることができる。 $g_h^{k,i}$ の通る単純路 $q_h^{k,i}$ は $A_j, j=0, 1, \dots, \ell-1$ に属する路を一つずつ含む。費用係数は $a \in A_0$ ならば $c^{\ell}(a)=1$, $a \in A^{\ell}-A_0$ ならば $c^{\ell}(a)=0$ なので、 $g_h^{k,i}$ による費用は $g_h^{k,i}$ の流量 v_h^k に等しい。よって g による総費用は

$$\sum_{k \in K} \sum_{h=1}^{H(k)} \sum_{i=0}^{\ell-1} v_h^k = \ell \sum_{k \in K} \sum_{h=1}^{H(k)} v_h^k = \ell \sum_{k \in K} d(k) = \ell D$$

となり、補題3.2より最適解である。(必要性の証明了)

次に十分性、即ち「CMCFの問題例 N^{ℓ} の最適解の総費用が ℓD であるならば、MCFLの問題例 (N, ℓ) が解を持つ」ことを示す。

N^{ℓ} の最適解の総費用が ℓD であると仮定する。補題3.1より単純流 g_1, g_2, \dots, g_H に分解できる最適解 g が存在する。 $h=1, 2, \dots, H$ について、 g_h の通る路を p_h 、流量を v_h 、経由枝数を ℓ_h とする。補題3.2の証明と同じ議論で、 $\ell_h = m_h \ell$ と書け、 p_h の作る費用は $m_h v_h$ である、但し m_h は自然数である。 f の総費用が ℓD と

なることから、

$$\sum_{h=1}^H m_h v_h = \ell D = \sum_{h=1}^H v_h$$

となり、この等式を満足する m_h は、 $m_h=1, h=1, 2, \dots, H$ に限られる。よって、 $\ell_h=\ell$ となり、 g_h の経路枝数は ℓ 、即ち g の経路枝数は ℓ 以下である。次に g に基づいて N 上の流れ f を以下の様に定義する。

$$f(a(x,y),k) = \sum_{i=0}^{\ell-1} \sum_{j=0}^{\ell-1} g(a(x_i, y_{(i+1) \bmod \ell}), k_j) / \ell, \quad a(x,y) \in A, \quad k \in K \quad (3.8)$$

また、

$$f'(a(x,y),k) = \sum_{i=0}^{\ell-1} \sum_{j=0}^{\ell-1} g(a(x_i, y_{(i+1) \bmod \ell}), k_j), \quad a(x,y) \in A, \quad k \in K$$

(すなわち、 $f(a,k) = \ell \cdot f'(a,k), a \in A, k \in K$)

とする。 f と f' の経路枝数は共に ℓ 以下である。さらに、 N 上の各枝の容量と各品種の要求量を ℓ 倍して得たネットワークを N' とすると、明かに f' は N' の実行可能解である。よって f は N の実行可能解であり、経路枝数は ℓ 以下である。(十分性の証明了) (定理3.1の証明了)

定理3.1より、経路枝数制限付き多品種流問題MCFLが最小費用多品種流問題CMCFに帰着されることが示された。これを利用して、MCFLを解く多項式時間アルゴリズムを得る。

【定理3.2】 MCFLはクラスPに属する。□

証明) まず、経路枝数の定義から、 $\ell \geq |V|$ ならば、任意の実行可能流の経路枝数が ℓ 以下になる。経路枝数制限を付けない多品種流問題は、CMCFで費用係数を $c(a)=0, a \in A$ としたものに等しく、クラスPに属する。よって $\ell < |V|$ の場合のみに話を限定して良い。MCFLの問題例 (N, ℓ) を多項式時間で解くアルゴリズムTORUSを以下に示す。

procedure TORUS

comment (N, ℓ): 入力の問題例

f. 出力される実行可能解 (存在するときのみ出力する);

begin

- 1 N に基づいて円環ネットワーク N' を構成する。
- 2 CMCFの問題例 N' を解き、最適解 g とその目標関数値 c_0 を得る。
- 3 if $c_0 > \ell D$ then "実行不可能"を出力;
else
- 4 (3.8)に従って、 g から、 f を構成する;

end;

本手続きがMCFLを解くアルゴリズムであることは、定理3.1によって保証される。次に多項式時間アルゴリズムであることを示す。 $|V|=O(\ell|V|)$, $|A'|=O(\ell(|V|+|A|))$, $|K'|=O(\ell|K|)$ なので、CMCFの問題例 N' を表現するのに必要なデータ量 $\text{length}(N')$ は、MCFLの問題例 (N, ℓ) を表現するのに必要なデータ量を $\text{length}(N, \ell)$ とすると、 $\text{length}(N')=O(\ell \text{length}(N, \ell))$ であり、第1行の円環化に要する時間量は $O(\ell \text{length}(N, \ell))$ である。第2行の操作は、CMCFがクラスPに属することから、 $\text{length}(N')$ の多項式時間、すなわち $\ell \text{length}(N, \ell)$ の多項式時間で可能。第4行は $O(|A'| |K'|)=O(\ell^2(|V|+|A|)|K|)$ 時間で出来る。ここで、 $\ell < |V|$ であることを考慮すれば、以上から、本アルゴリズムが多項式時間アルゴリズムであることが分かる。よってMCFLはクラスPに属する。 Q.E.D.

3. 3. 均等配分制約の付いた問題

通常が多品種流問題(MCF)では、各路を流れる流量は、流量保存則と容量制約を満たす限り、自由に設定できる。しかし通信網への適用を考える場合、一つの品種を流すことのできる路の集合は確定的に指示できるが、各路にどれだけの割合で流れを配分するかは、確率的に変動し制御できない場合がある。例えば通信網における接続経路の候補群を決定する問題[ITO 92c] [ITO 92d]は、各品種の流路の集まりを決定することを目的とし、各接続経路に流す量は各交換機が網状態に応じて路を決定するアルゴリズムに依存する。最も単純かつ重要なアルゴリズムとして、各交換機が無作為に路を決定する場合を考えると、この場合は各路への流れの配分は均等になる。そこで、本節では、制約付き多品種流問題の一つとして、品種毎に各路へ均等に流れを配分することを前提として、各品種を流す路の集合を求める問題、多品種流路問題(MCR: Multi-Commodity flow Routes problem)を考える[ITO 92a] [ITO 94d]。

MCRは「独立経路問題」(disjoint paths problem)[IBAR 89]の一般化に近い問題である。独立経路問題の詳しい定義は省略するが、一般に難しい問題であり、グラフの形状や入力値に制限を加えたある種の問題についてのみクラスPに属することが指摘されている[KARP 72] [EVEN 76] [PERL 78]。従って独立経路問題からの類推で、MCRが難しい問題であろうとの予想がつくが、理論的に示した例は無い。また経路枝数や品種数に制約を加えた場合の計算量についても論じられていない。本節ではこれらのことを明らかにする。すなわち3. 3. 1で諸定義を行なった後、3. 3. 2で本問題が、品種数を2以上の任意の整数に固定した場合でも強NP完全であり、品種数が1の場合のみ多項式時間アルゴリズムが存在する[ITO 94d]ことを示す。また、通信網における接続経路は、長くなると品質の劣化や制御上の困難を伴うので、中継交換機を1つ、つまりグラフ上の表記では、経由する枝数を2以下に限定する場合が多い[GIMP 74] [ASH 92] [INOUE 92] [ITO 92c]。そこで3. 3. 3では経路枝数を2に限定した場合のMCR (MCR2)を扱い、それでもなお強NP完全であることを示す。ただし、MCR2の場合は、品種数を任意の定数に固定すると、多項式時間アルゴリズムが存在する[ITO 94d]。

3. 3. 1. 多品種流路問題

ネットワーク N に関する定義は3. 2. 1 参照。但し、本節では費用は使用しないので、ネットワークを表記する際には、 $N=(G,K,S,d,u)$ の様に費用係数を省略する。なお、本節(3. 3)で扱うグラフ G は単純な有向グラフに限るものとする。

【定義】

P : グラフ G 上の路集合

$g(p,k) \in \{0 \text{ 以上の実数}\}$: 路 $p \in P$ 上の品種 $k \in K$ の流量。

枝 $a \in A$ を含む路の集合を $F(a) = \{p \in P \mid a \in A(p)\}$ と記す。

従って、 $f(a,k) = \sum_{p \in F(a)} g(p,k)$ の性質が成立する。□

【定義】 多品種流路問題(MCR)

[入力] ネットワーク $N=(G,K,S,d,u)$

[要請] 下記の条件(i)~(v)を満足する流路集合 $Q(k) \subseteq P, k \in K$ が存在するか否かを求めよ。

(i) 各 $p \in Q(k)$ の始点は $s(k)$ であり、終点は $t(k)$ 。

(ii) 流量保存則:

$$\sum_{p \in Q(k)} g(p,k) = d(k), k \in K$$

$$\sum_{p \in P - Q(k)} g(p,k) = 0, k \in K$$

(iii) 容量制約:

$$(0 \leq) F(a) \leq u(a), a \in A$$

但し、 $F(a) = \sum_{k \in K} f(a,k)$ 、 $f(a,k) = \sum_{p \in F(a)} g(p,k)$ である。

(iv) 流量配分則: 任意の $k \in K$ に対し

$$g(p,k) = g(p',k), p, p' \in Q(k)$$

(v) $p, p' \in Q(k)$ かつ $p \neq p'$ ならば p と p' は内素。□

条件(iv)と(v)が無ければ、MCRは通常が多品種流問題MCFになる。

3. 3. 2. 多品種流路問題～経路枝数制限の無い問題

本節では、多品種流路問題MCRの計算量を、品種数を変化させて考える。まず、1品種の問題は、以下に述べる様に2点間の内素な路の最大数を求める問題に帰着することができ、多項式時間のアルゴリズムが存在する。

【定理3.3】 品種数 $|K|$ が1に固定されたMCRはクラスPに属する。□

証明) 品種数が1のMCRの問題例 $N=(G,K,S,d,u)$ が与えられたとする。ただし、

$$s=s(k_1), t=t(k_1), d=d(k_1), Q=Q(k_1)$$

と記す。 s の出次数及び t の入次数を $\deg_{out}(s), \deg_{in}(t)$ とすると、MCRの条件(v)より、

$$|Q| \leq \min(\deg_{out}(s), \deg_{in}(t)) \leq n-1$$

である。そこで $|Q|$ を1から $\min(\deg_{out}(s), \deg_{in}(t))$ まで変化させて、それぞれについて解が存在するかどうかを調べる。そのため、 $|Q|$ を q に固定した問題を考える。

条件(iv)より

$$g(p, k_1) = d/q, p \in Q$$

である。よってグラフ G の枝のうち、容量が d/q より小さい枝を除去したグラフ $G(q)$ に対して、 s, t 間の内素な路の最大数 $M(q)$ を求めれば良い。 $|Q|=q$ に固定した問題が解を持つ必要十分条件は $M(q) \geq q$ である。 $M(q)$ を求める問題はMengerの定理[MENG 27] [HARA 69]より、 $G(q)$ における s, t 間の点連結度 $\kappa(s, t, G(q))$ を求めることに等しく、最大流問題に帰着することによって多項式時間で解くことが出来る[DINI 78] [KING 92]。

よって品種数が1のMCRは s, t 間の点連結度を求める問題を高々 $n-1$ 度解くことによって、解くことができる。故にクラスPに属する。 Q.E.D.

しかし、MCRにおいて品種数が2になると既に強NP完全になる。

【定理3.4】 品種数が2に固定されたMCRは強NP完全である。□

定理3.4の証明は[EVEN 76]の2品種整数流問題(Two-commodity integral

flow in directed graphs: D2CIF)の証明と全く同様にして、充足可能性問題を2品種のMCRへ帰着することによって証明できる。D2CIFは問題の制約としてMCRの条件(iv)の均等配分則、及び条件(v)の内素性は保有していない。しかし、充足可能性問題を帰着して出来たD2CIFの問題例は、(iv)と(v)の性質を偶然にも満たすので、そのまま品種数2のMCRのNP完全性の証明とすることが出来る。しかも、同様の帰着法で出来た品種数2のMCRの問題例においては、枝の容量は全て1、需要量は高々 $O(|V|)$ となる。よってこの結果は、品種数2のMCRの強NP完全性も意味する。証明の概要を付録2に掲げる。

この結果、2以上の任意の固定された品種数に対して、MCRが強NP完全であることが容易に分かる。

3. 3. 3. 経路枝数制限のある問題

前節では、多品種流路問題(MCR)がNP困難であることを示した。これに対し、本章では、経路枝数を2以下に限ったMCRの複雑さを考察し、一般のMCRとの相違点を明らかにする。

【定義】 MCRの定義において、各 $Q(k)$, $k \in K$ を、2つ以下の枝から構成される路の集合 P_2 の中から選ぶように制約を付けた問題を、経路枝数2以下の多品種流路問題と呼び、MCR2と記す。□

MCR2はMCRに比べ、解の範囲が狭くなっているので、直感的には簡単になっている様に思える。しかし、下記の定理に示す様に、やはり強NP完全である。

【定理3.5】 MCR2は強NP完全である。□

定理3.5は、既知の強NP完全問題である3分割問題(3PRT)[GARE 79]をMCR2へ帰着することによって証明できる。3分割問題の定義を記す。

【3分割問題(3PRT)】

[入力] 有限集合 $X=\{x_1, x_2, \dots, x_{3m}\}$ (ただし m は自然数)

要素の大きさ $z(x) \in \{\text{自然数}\} (x \in X)$, 但し $z(x)$ は次条件を満足する。

$$B/4 < z(x) < B/2, x \in X$$

$$\sum_{x \in X} z(x) = mB$$

[要請] $\sum_{x \in X_i} z(x) = B$ となる様に、 X は部分集合 X_1, X_2, \dots, X_m に分割可能か。(このとき、上記の制約から各 X_i の位数は3となる。) \square

定理3.5の証明)

MCR2がNPに属することは明らかである。よって以下、3PRTをMCR2に帰着することで、MCR2の強NP困難性を示す。3PRTの任意の問題例 (X, z) に対し、MCR2の問題例 $N(X, z)$ を以下のように作成する (図3.3参照)。

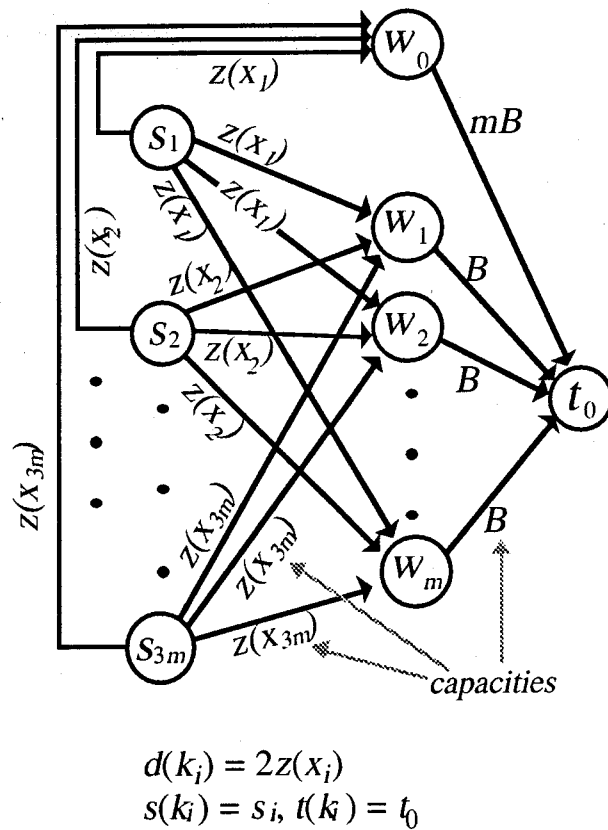


図3.3 3PRTのMCRへの帰着法

Fig. 3.3. Transformation of 3PRT to MCR.

$$N(X, z) = (G, K, S, d, u)$$

$$G = (V, A)$$

$$V = V_s \cup V_w \cup \{t_0\}$$

$$V_s = \{s_1, s_2, \dots, s_{3m}\}$$

$$V_w = \{w_0, w_1, w_2, \dots, w_m\}$$

$$A = A_s \cup A_t$$

$$A_s = \{a(s_i, w_j) \mid s_i \in V_s, w_j \in V_w\}$$

$$A_t = \{a(w_j, t_0) \mid w_j \in V_w\}$$

$$K = \{k_1, k_2, \dots, k_{3m}\}$$

$$S = \{(s(k_i) = s_i, t(k_i) = t_0) \mid k_i \in K\}$$

$$d(k_i) = 2z(x_i), k_i \in K$$

$$u(a(s_i, w_j)) = z(x_i), s_i \in V_s, w_j \in V_w$$

$$u(a(w_0, t_0)) = mB$$

$$u(a(w_i, t_0)) = B, i = 1, 2, \dots, m$$

このとき、「3PRTの問題例 (X, z) に解が存在する」の必要十分条件が「MCR2の問題例 $N(X, z)$ に解が存在する」であることを示せば良い。

(必要性) 「3PRTの問題例 (X, z) に解が存在する」ならば「MCR2の問題例 $N(X, z)$ に解が存在する」ことの証明。

(X, z) の解を $X_j^* (j=1, 2, \dots, m)$ とする。

$$Q^*(k_j) = \{\langle s_i, w_j, t_0 \rangle, \langle s_i, w_0, t_0 \rangle \mid x_i \in X_j^*\}$$

とすると、条件(i)と(v)を満たす。条件(ii)と(iv)を満たすように配分すると、

$$g(\langle s_i, w_j, t_0 \rangle, k_j) = g(\langle s_i, w_0, t_0 \rangle, k_j) = z(x_i) \quad (\text{ただし } x_i \in X_j^*)$$

$$g(\langle s_i, w, t_0 \rangle, k_j) = 0 \quad (\text{ただし } w \in V_w - \{w_0, w_j\}, x_i \in X_j^*)$$

となる。よって、

$$F(a(s_i, w)) \leq z(x_i), s_i \in V_s, w \in V_w \quad (3.9)$$

また、

$$F(a(w_0, t_0)) = \sum_{x_i \in X} z(x_i) = mB \quad (3.10)$$

$$F(a(w_j, t_0)) = \sum_{x_i \in X_j^*} z(x_i) = B, j = 1, 2, \dots, m \quad (3.11)$$

が成立する。式(3.9)～(3.11)より条件(iii)を満たす。

以上により、 $Q^*(k_j)$ は条件(i)～(v)を全て満たし、 $N(X, z)$ の解となっている。

(必要性の証明了)

(十分性) MCR2の問題例 $N(X, z)$ に解が存在する」ならば「3PRTの問題例 (X, z) に解が存在する」ことの証明。

$N(X, z)$ の解を $Q^*(k_i), k_i \in K$ とする。まず、

$$\sum_{k_i \in K} d(k_i) = 2mB$$

なので、カット条件より A_i の枝は飽和して（上限まで使用されて）いなければならない。次に、

$$\sum_{s_i \in V_S} u(a(s_i, w_0)) = mB = u(a(w_0, t_0))$$

より、 $a(s_i, w_0), i=1, 2, \dots, 3m$ も飽和している。よって条件(ii)を考慮すると、 $Q^*(k_i)$ は $\langle s_i, w_0, t_0 \rangle$ を含む2つの路から構成されている。 $\langle s_i, w_0, t_0 \rangle$ 以外の $Q^*(k_i)$ の要素（路）を r_i とおく。 A_i の枝が飽和していることから、各 $a(w_j, t_0), j=1, 2, \dots, m$ について

$$\sum_{\langle s_i, w_j, t_0 \rangle = r_i} g(r_i, k_i) = F(a(w_j, t_0)) = B \quad (3.12)$$

が成立する。ここで

$$X_j^* = \{x_i \mid \langle s_i, w_j, t_0 \rangle = r_i\}$$

とおくと、(3.12)より

$$\sum_{x \in X_j^*} z(x) = \sum_{\langle s_i, w_j, t_0 \rangle = r_i} g(r_i, k_i) = B \quad (3.13)$$

となる。よって、 $X_j^*, j=1, 2, \dots, m$ は (X, z) の解になっている。（十分性の証明完了）

以上の結果必要十分性が示された。（定理3.5の証明完了）

一般のMCRは品種数が2であっても強NP完全であった（定理3.4）。しかしMCR2は品種数 $|K|$ が定数値の場合、以下に示す様に多項式時間アルゴリズムによって解くことができる。

【定義】 品種数 $|K|$ を定数 b に固定したMCR2のことを、MCR2- b と記す。□

【定理3.6】 任意の自然数 b に対し、MCR2- b はクラス P に属する。□

以下、いくつかの補題を経てこの定理を証明していく。まずMCR2- b の問題例 $N=(G=(V, A), K, S, d, u)$ が与えられたら、前処理として、枝の存在しない節点 $v, w \in V$ 間に容量0の枝を付与し、グラフ $G=(V, A)$ を完全グラフにする（明らかに同じ問題である）。

【定義】 $V_s := \{s(k) \mid k \in K\}$, $V_t := \{t(k) \mid k \in K\}$

さらに $V_s := \{s_1, s_2, \dots, s_{|V_s|}\}$ と記し、 $K_s(j) := \{k \in K \mid s_j = s(k)\}$ とおく。

同様に $V_t := \{t_1, t_2, \dots, t_{|V_t|}\}$, $K_t(j) := \{k \in K \mid t_j = t(k)\}$ を定義する。□

【補題3.3】 各品種 $k \in K$ 毎に 0 以上の整数 $p(k), q(k)$ を入力変数として持ち、MCR2- b に「品種 k の解の路の集合 $\mathcal{Q}(k)$ のうち、 $\langle s(k), v, t(k) \rangle$, $v \in V_s \cup V_t$ または $\langle s(k), t(k) \rangle$ なる路が $p(k)$ 個で、 $\langle s(k), v, t(k) \rangle$, $v \in V - (V_s \cup V_t)$ なる路が $q(k)$ 個でなければならない」という制約を付け加えた問題 (MCR2 α - b と呼ぶことにする) がクラス P に属するならば、MCR2- b もクラス P に属する。□

補題3.3の証明) $p(k), q(k)$ は、 $0 \leq p(k), q(k) \leq |V|-1$ の整数なので、各品種 $k \in K$ 毎に $|V|^2$ 通りずつ考えられる。よって MCR2 α - b を全部で高々 $|V|^{2b}$ 回解くことによって、MCR2- b を解くことができる。 $b=|K|$ が定数なので、これは $|V|$ の多項式である。よって MCR2 α - b が P に属するならば MCR2- b も P に属する。Q.E.D.

【定義】 各品種 $k \in K$ 毎に 0 以上の整数 $p(k)$ を入力変数として持ち、MCR2- b に「品種 k の解の路の集合 $\mathcal{Q}(k)$ は、 $\langle s(k), v, t(k) \rangle$, $v \in V_s \cup V_t$ または $\langle s(k), t(k) \rangle$ なる路 $p(k)$ 個のみから構成されなければならない」という制約を付け加えた問題を MCR2 β - b と呼ぶ。また、各品種 k 毎に 0 以上の整数 $q(k)$ を入力変数として持ち、MCR2- b に「品種 k の解の路の集合 $\mathcal{Q}(k)$ は $\langle s(k), v, t(k) \rangle$, $v \in V - (V_s \cup V_t)$ なる路 $q(k)$ 個のみから構成されなければならない」という制約を付け加えた問題を MCR2 γ - b と呼ぶ。□

【補題3.4】 MCR2 β - b 及び MCR2 γ - b がクラス P に属するならば、MCR2 α - b もクラス P に属する。□

証明) 均等配分則より、MCR2 α - b では、選択された路に配分される品種 $k \in K$ の流量 $\delta(k)$ は

$$\delta(k) = d(k) / (p(k) + q(k))$$

の様に固定される。また、 $\langle s(k), t(k) \rangle$, $\langle s(k), v, t(k) \rangle$, $v \in V_s \cup V_t$ と $\langle s(k'), v, t(k') \rangle$, $v \in V - (V_s \cup V_t)$ は共通の枝を使用しない、すなわち

$$\{a(s(k), t(k)), a(s(k), v), a(v, t(k)) \mid k \in K, v \in V_s \cup V_t\}$$

$$\cap \{a(s(k), v), a(v, t(k)) \mid k \in K, v \in V - (V_s \cup V_t)\} = \phi$$

であるから（図3.4参照）、MCR2 β - b の品種 $k \in K$ の要求量 $d'(k)$ を

$$d'(k) = d(k)p(k)/(p(k) + q(k))$$

MCR2 γ - b の品種 $k \in K$ の要求量 $d''(k)$ を

$$d''(k) = d(k)q(k)/(p(k) + q(k))$$

とにおいて、MCR2 β - b 及びMCR2 γ - b を独立に解くことによって、MCR2 α - b の解とすることが出来る。以上から題意が証明される。Q.E.D.

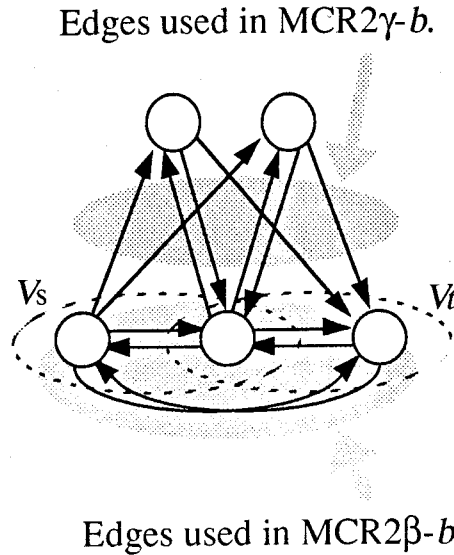


図3.4 MCR2 β - b とMCR2 γ - b の使用する枝

Fig. 3.4 Edges used in MCR2 β - b and MCR2 γ - b .

以上から、次の補題が直ちに導かれる。

【補題3.5】 MCR2 β - b 及びMCR2 γ - b がクラスPに属するならば、MCR2- b もクラスPに属する。□

証明) 補題3.3、3.4より明か。Q.E.D.

補題3.5より、MCR2 β - b 及びMCR2 γ - b がクラスPに属することを示せば定理3.6は証明される。まずMCR2 β - b から考える。

【補題3.6】 MCR2 β - b はクラスPに属する。□

補題3.6の証明) MCR2 β - b は、対象とするグラフを $V_s \cup V_t$ とその間の枝に限れば良い。よって扱うグラフは $G_\beta = (V_\beta, A_\beta)$, $V_\beta = V_s \cup V_t$, $A_\beta = \{a(v, w) \in A \mid v, w \in V_\beta\}$ として良い。 $|V_\beta| \leq 2|K| = 2b$ より、各品種 $k \in K$ について、 $Q(k)$ の選び方は ${}_{2b}C_{p(k)}$ ($\leq 2^{2b}$) 以下である。よって、全ての品種 k について $Q(k)$ を選ぶ選び方は $b \cdot 2^{2b}$ 以下。選んだ $Q(k)$, $k \in K$ が問題の条件を満たすことは、 $O(|A_\beta| + \sum_{k \in K} p(k))$ 時間で判定出来る。ここで、 $|A_\beta| \leq |V_\beta|^2 \leq 4b^2$ 及び $p(k) \leq |V_\beta| \leq 2b$ を考慮すると、 $O(|A_\beta| + \sum_{k \in K} p(k)) \leq O(b^2)$ であり、これは定数オーダーである。よって、MCR2 β - b は定数時間アルゴリズムで解くことができる。故にクラスPに属する。 Q.E.D.

よって以下、MCR2 γ - b がクラスPに属することを示せば良い。

MCR2 γ - b では、グラフの枝は

$$\{a(s_j, v), a(v, t_j) \in A \mid s_j \in V_s, t_j \in V_t, v \in V - V_s \cup V_t\}$$

のみ使用する。よって本章ではこれ以降、グラフの枝集合 A はこの様なもののみに限ることとする。解かりやすくするために、図3.5の問題例を使つての説明も添える。

【定義】 $B \in 2^K$ (2^K : K の部分集合の集合) に対して、その重さを

$\omega(B) = \sum_{k \in B} \delta(k)$ と定める。また、全ての $B \in 2^K$ に $\omega(B)$ の小さな順 (同じ大きさのものは任意の順) に番号 $1, 2, \dots, |2^K| = 2^b$ を対応させ、 $\text{ord}(B)$ で表す。

また $\text{ord}(B) = h$ の逆関数を $\text{ord}^{-1}(h) = B$ とする。□

つまり、 $\text{ord}(B) < \text{ord}(B')$ ならば $\omega(B) \leq \omega(B')$ の性質がある。当然 $\text{ord}(\phi) = 1$, $\text{ord}(K) = |2^K| = 2^b$ となる。図3.5の例では $\omega(\phi) = 0$, $\omega(\{k_1\}) = 2$, $\omega(\{k_2\}) = 3$, $\omega(\{k_1, k_2\}) = 5$ であり、従つて、 $\text{ord}(\phi) = 1$, $\text{ord}(\{k_1\}) = 2$, $\text{ord}(\{k_2\}) = 3$, $\text{ord}(\{k_1, k_2\}) = 4$ となる。

任意の品種 k に対して、 $Q(k)$ 内の異なる路 p, p' は互いに内素なので、枝 $a \in A$ を通る品種 $k \in K$ の流量は $\delta(k)$ または 0 となる。よって次に定義する様

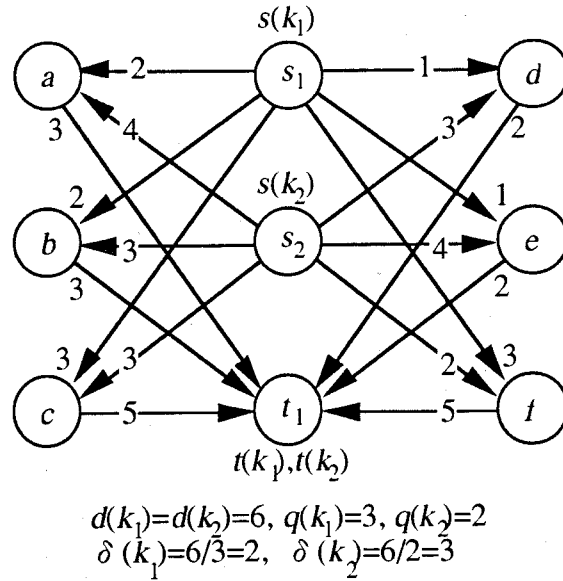


図3.5 定理3.6の証明の説明用の問題例

Fig. 3.5. An example for the proof of theorem 3.6.

に、枝集合 A をその容量によって $|2^k|=2^b$ (定数) 個の部分集合に分割する。

【定義】

$$A_1(h) := \{a \in A \mid \omega(\text{ord}^{-1}(h)) \leq u(a) < \omega(\text{ord}^{-1}(h+1))\}, h=1, 2, \dots, 2^b-1$$

$$A_1(2^b) := \{a \in A \mid \omega(\text{ord}^{-1}(2^b)) \leq u(a)\} \quad \square$$

図3.5の問題例の A_1 による分割結果を図3.6に示す。 $a \in A_1(h)$ の意味は、枝 a に配分する品種の集合を B とすると、 $\text{ord}(B) \leq h$ でなければならず、また $\text{ord}(B) \leq h$ であれば $a \in A_1(h)$ に品種の集合 B を配分しても容量上問題が無いことを表す。

【定義】 $V-(V_s \cup V_t)$ の分割 $V_2(h^s, h^t)$ を以下のように定める。但し $h^s=(h_1^s, h_2^s, \dots, h_{|V_s|}^s)$ 、 $h^t=(h_1^t, h_2^t, \dots, h_{|V_t|}^t)$ かつ $h_k^s, h_k^t \in \{1, 2, \dots, 2^b\}$ である。

$v \in V-(V_s \cup V_t)$ が $v \in V_2(h^s, h^t)$ となる必要十分条件は、 $a(s_j, v) \in A_1(h_j^s)$, $\forall j \in \{1, 2, \dots, |V_s|\}$ かつ $a(v, t_j) \in A_1(h_j^t)$, $\forall j \in \{1, 2, \dots, |V_t|\}$ である。各 v が属する $V_2(h^s, h^t)$ は唯一つ存在する。□

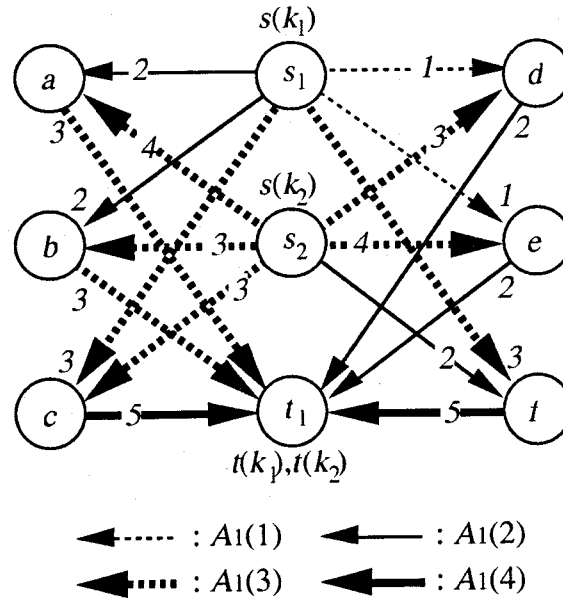


図3.6 A_1 の例

Fig. 3.6 An example for A_1 .

【定義】 A の分割 $A_2(j, \ell, h^s, h')$ を以下のように定める。但し前述の様に $h^s = (h_1^s, h_2^s, \dots, h_{|V_s|}^s)$ 、 $h' = (h_1', h_2', \dots, h_{|V_t|}')$ かつ $h_k^s, h_k' \in \{1, 2, \dots, 2^b\}$ であり、さらに $j \in \{1, 2, \dots, \max(|V_s|, |V_t|)\}$ 、 $\ell \in \{1, 2\}$ である。

(1) $a \in \{a(s_j, v) \mid s_j \in V_s, v \in V - (V_s \cup V_t)\}$ の場合。

$v \in V_2(h^s, h')$ ならば、 $a \in A_2(j, 1, h^s, h')$ 。

(2) $a \in \{a(v, t_j) \mid t_j \in V_t, v \in V - (V_s \cup V_t)\}$ の場合。

$v \in V_2(h^s, h')$ ならば、 $a \in A_2(j, 2, h^s, h')$ 。□

図3.5の問題例の V_2 による分割結果を図3.7に、 A_2 による分割結果を表3.1に示す。 A_2 は高々 $b \cdot 2 \cdot (2^b)^b \cdot (2^b)^b = 2b \cdot 2^{2b^2}$ (定数) 個の集合への分割である (但し $a^b := a^b$ とする)。

次にMCR2 γ - b の解法を作成するために、 A_2 と V_2 を用いて変数を定義する。

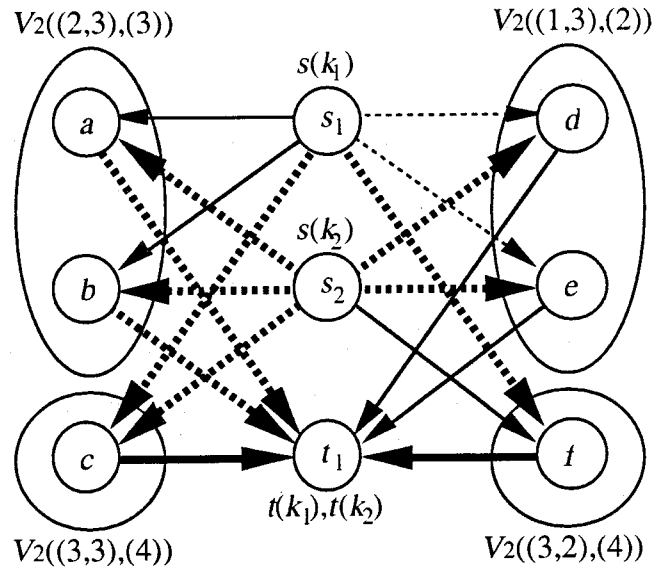


図3.7 V_2 の例

Fig. 3.7 An example for V_2 .

表3.1 A_2 の例

Tab. 3.1. An example for A_2 .

| | $a(s_1,*)$ | $a(s_2,*)$ | $a(*,t_1)$ |
|-----|-------------------|-------------------|-------------------|
| a | $(1,1,(2,3),(3))$ | $(2,1,(2,3),(3))$ | $(1,2,(2,3),(3))$ |
| b | $(1,1,(2,3),(3))$ | $(2,1,(2,3),(3))$ | $(1,2,(2,3),(3))$ |
| c | $(1,1,(3,3),(4))$ | $(2,1,(3,3),(4))$ | $(1,2,(3,3),(4))$ |
| d | $(1,1,(1,3),(2))$ | $(2,1,(1,3),(2))$ | $(1,2,(1,3),(2))$ |
| e | $(1,1,(1,3),(2))$ | $(2,1,(1,3),(2))$ | $(1,2,(1,3),(2))$ |
| f | $(1,1,(3,2),(4))$ | $(2,1,(3,2),(4))$ | $(1,2,(3,2),(4))$ |

$(*,*,(*,*),(*))$ means $A_2(*,*,(*,*),(*))$.

【定義3.1】

$x(k, j, \ell, h^s, h^t) \in \{0 \text{ 以上の整数}\}$: 品種 k が流路として選択した路に属する枝のうち、 $A_2(j, \ell, h^s, h^t)$ に属する枝の数。

$y(h, j, \ell, h^s, h^t) \in \{0 \text{ 以上の整数}\}$: 全ての $k \in \text{ord}^{-1}(h)$ が流路として選択し、かつどの $k \in K - \text{ord}^{-1}(h)$ も流路として選択しなかった路に属する枝のうち、 $A_2(j, \ell, h^s, h^t)$ に属する枝の数。

$z(\eta^s, \eta^t, h^s, h^t) \in \{0 \text{ 以上の整数}\}$: 全ての $j \in \{1, 2, \dots, |V_s|\}$ に対して「全ての $k \in \text{ord}^{-1}(\eta_j^s)$ が $a(s, v)$ を流路として選択し、かつどの $k \in K - \text{ord}^{-1}(\eta_j^s)$ も $a(s, v)$ を流路として選択せず」かつ、全ての $j \in \{1, 2, \dots, |V_t|\}$ に対して「全ての $k \in \text{ord}^{-1}(\eta_j^t)$ が $a(v, t_j)$ を流路として選択し、かつどの $k \in K - \text{ord}^{-1}(\eta_j^t)$ も $a(v, t_j)$ を流路として選択しない」という条件を満たす節点 $v \in V_2(h^s, h^t)$ の数。但し、 $h \in \{1, 2, \dots, 2^b\}$, $h^s, \eta^s \in \{1, 2, \dots, 2^b\}^{|V_s|}$, $h^t, \eta^t \in \{1, 2, \dots, 2^b\}^{|V_t|}$, $\ell \in \{1, 2\}$ 。

□

これまで定義した記号を用いて、MCR2 γ - b の解となる必要条件として制約式<3.1>~<3.11>を定義する。但し、記号「 $\Sigma \langle : \rangle$ 」は「 $\langle \rangle$ 」内の「 $:$ 」の左側の変数について、「 $:$ 」の右側の制約を満たす範囲で、全ての組合わせに対しての和を表す。また式後半の記号「 $(:)$ 」は「 $()$ 」内の「 $:$ 」の左側の変数について、「 $:$ 」の右側の制約を満たす範囲で、全ての組合わせに対して制約式が存在することを示す。「 $:$ 」の無い「 $\Sigma \langle \rangle$ 」や「 $()$ 」は、制約なしの全ての組合せという意味である。

$$z(\eta^s, \eta^t, h^s, h^t) = 0, (\eta^s, \eta^t, h^s, h^t : \cup_j \text{ord}^{-1}(\eta_j^s) \neq \cup_j \text{ord}^{-1}(\eta_j^t)) \quad <3.1>$$

$$\Sigma \langle h^s, h^t \rangle x(k, j, \ell, h^s, h^t) = q(k), (k, j, \ell : k \in K_s(j)) \quad <3.2>$$

$$x(k, j, 1, h^s, h^t) = 0, (k, j, h^s, h^t : k \in K - K_s(j)) \quad <3.3>$$

$$x(k, j, 2, h^s, h^t) = 0, (k, j, h^s, h^t : k \in K - K_t(j)) \quad <3.4>$$

$$\Sigma \langle h \rangle y(h, j, \ell, h^s, h^t) = |A_2(j, \ell, h^s, h^t)|, (j, \ell, h^s, h^t) \quad <3.5>$$

$$y(h, j, 1, h^s, h^t) = 0, (h, j, \ell, h^s, h^t : h > h_j^s) \quad <3.6>$$

$$y(h, j, 2, h^s, h^t) = 0, (h, j, \ell, h^s, h^t : h > h_j^t) \quad <3.7>$$

$$\Sigma \langle \eta^s, \eta^t \rangle z(\eta^s, \eta^t, h^s, h^t) = |V_2(h^s, h^t)|, (h^s, h^t) \quad <3.8>$$

$$x(k, j, \ell, h^s, h^t) = \Sigma \langle h : k \in \text{ord}^{-1}(h) \rangle y(h, j, \ell, h^s, h^t), (j, \ell, h^s, h^t) \quad <3.9>$$

$$y(h, j, 1, h^s, h^t) = \Sigma \langle \eta^s, \eta^t : \eta_j^s = h \rangle z(\eta^s, \eta^t, h^s, h^t), (h, j, h^s, h^t) \quad <3.10>$$

$$y(h, j, 2, h^s, h^t) = \Sigma \langle \eta^s, \eta^t : \eta_j^t = h \rangle z(\eta^s, \eta^t, h^s, h^t), (h, j, h^s, h^t) \quad <3.11>$$

制約式<3.1>～<3.11>の意味を簡単に説明する。

制約式<3.1>は $V-(V_s \cup V_t)$ における流路数の保存則である。制約式<3.2>は各品種 k の流路が $q(k)$ になっていることを表す。制約式<3.3>, <3.4>は品種毎の流出点、流入点のどちらとも接続していない枝は流路に選ばれないことを表す。制約式<3.5>は使用する枝数の上限制約である。不等式ではなく等式になっている理由は、どの品種にも流路として選ばれなかった枝の本数を表す変数 $y(1, j, l, h^s, h^t)$ が存在するからである。制約式<3.6>, <3.7>は枝の容量制約である。制約式<3.8>は使用中継節点 ($V-(V_s \cup V_t)$) 数の上限制約である。やはり<3.5>と同じ理由で、不等式とならずに等式になっている。制約式<3.9>は x と y が整合していることを表す。制約式<3.10>, <3.11>は y と z が整合していることを表す。

【補題3.7】 $\text{MCR2}\gamma\text{-}b$ に解が存在する必要十分条件は、対応する制約式<3.1>～<3.11>を満足する x, y, z が存在することである。□

証明)

(必要条件であることの証明)

$\text{MCR2}\gamma\text{-}b$ の解 $Q(k)$ が存在したと仮定する。 $Q(k)$ に基づいて x, y, z を構成する(定義3.1参照)。制約式<3.1>～<3.11>は $Q(k)$ の必要条件を列挙したものであるから、これらを満足することは明らかである。(必要性の証明了)

(十分条件であることの証明)

制約式<3.1>～<3.11>を満足する x, y, z が存在したと仮定する。それらを用いて「 $\text{MCR2}\gamma\text{-}b$ 」の解 $Q(k)$ を以下の手続きMAKEで作成する。


```

procedure MAKE
begin
1  for each  $k \in K$  do
2     $Q(k) := \phi$ 
    enddo;
3  for each  $h^s \in \{1, 2, \dots, 2^b\}^{|V^s|}$ ,  $h^t \in \{1, 2, \dots, 2^b\}^{|V^t|}$  do
4     $V_2(h^s, h^t)$  を  $|V_3(\eta^s, \eta^t, h^s, h^t)| = z(\eta^s, \eta^t, h^s, h^t)$  を満足するように、
       $V_3(\eta^s, \eta^t, h^s, h^t)$  に任意に分割する；
5    for each  $\eta^s \in \{1, 2, \dots, 2^b\}^{|V^s|}$ ,  $\eta^t \in \{1, 2, \dots, 2^b\}^{|V^t|}$  do
6      for each  $v \in V_3(\eta^s, \eta^t, h^s, h^t)$  do
7        for each  $k \in K$  do
8          if  $s_j = s(k)$ ,  $t_{j'} = t(k)$  なる  $j$  と  $j'$  に対し、 $k \in \text{ord}^{-1}(\eta_j^s)$  かつ  $k \in \text{ord}^{-1}(\eta_{j'}^t)$ 
9          then  $Q(k) := Q(k) \cup \{ \langle s(k), v, t(k) \rangle \}$  fi;
        enddo;
      enddo;
    enddo;
  enddo;
end;

```

第4行の分割は制約式<3.8>により可能であるので、手続きMAKEは実行可能である。

次に本手続きによって作成された $Q(k)$ が解となっていることを示す。
 $\langle s(k), v, t(k) \rangle \in Q(k)$ が $s(k), t(k)$ 間の経路枝数2の路となっていることは明らかなので、(イ) 流路数が $q(k)$ であることと (ロ) 各枝の容量制約を満たすことを示せば良い。

(イ) について。

$k \in K_s(j)$, $k \in K_t(j')$ とすると、

$$\begin{aligned}
 q(k) &= \sum \langle h^s, h^t \rangle x(k, j, 1, h^s, h^t) \quad (\text{制約式<3.2>}) \\
 &= \sum \langle h, h^s, h^t: k \in \text{ord}^{-1}(h) \rangle y(h, j, 1, h^s, h^t) \quad (\text{制約式<3.9>より}) \\
 &= \sum \langle \eta^s, \eta^t, h^s, h^t: k \in \text{ord}^{-1}(\eta_j^s) \rangle z(\eta^s, \eta^t, h^s, h^t) \quad (\text{制約式<3.10>より}) \\
 &= \sum \langle \eta^s, \eta^t, h^s, h^t: k \in \text{ord}^{-1}(\eta_j^s), k \in \text{ord}^{-1}(\eta_{j'}^t) \rangle z(\eta^s, \eta^t, h^s, h^t) \quad (\text{制約式<3.4>, <3.9>, <3.11>より、ある } j'' \neq j' \text{ について } k \in \text{ord}^{-1}(\eta_{j''}^t) \text{ ならば})
 \end{aligned}$$

$$\begin{aligned}
& z(\eta^s, \eta', h^s, h') = 0 \text{なので、制約式<3.1>より、} k \in \text{ord}^{-1}(\eta_j^s) \text{かつ} \\
& k \in K\text{-ord}^{-1}(\eta_j') \text{ならば } z(\eta^s, \eta', h^s, h') = 0 \text{となるので} \\
& = \sum \langle \eta^s, \eta', h^s, h' : k \in \text{ord}^{-1}(\eta_j^s), k \in \text{ord}^{-1}(\eta_j') \rangle |V_3(\eta^s, \eta', h^s, h')| \\
& = |\mathbb{Q}(k)|
\end{aligned}$$

よって (イ) が示された。

(ロ) について。

制約式<3.6>, <3.7>, <3.10>, <3.11>より

$z(\eta^s, \eta', h^s, h') > 0$ ならば、

$$\eta_j^s \leq h_j^s, j=1, 2, \dots, |V_s| \text{かつ } \eta_j' \leq h_j', j=1, 2, \dots, |V_l|$$

である。よって

$|V_3(\eta^s, \eta', h^s, h')| > 0$ ならば、

$$\eta_j^s \leq h_j^s, j=1, 2, \dots, |V_s| \text{かつ } \eta_j' \leq h_j', j=1, 2, \dots, |V_l|$$

となる。ところが、 $a(s_j, v)$, $v \in V_3(\eta^s, \eta', h^s, h')$ を $\mathbb{Q}(k)$ の一部として選ぶ品種集合は $\text{ord}^{-1}(\eta_j^s)$ である。同様に $a(v, t_j)$, $v \in V_3(\eta^s, \eta', h^s, h')$ を $\mathbb{Q}(k)$ の一部として選ぶ品種集合は $\text{ord}^{-1}(\eta_j')$ である。ここで、

$$v \in V_3(\eta^s, \eta', h^s, h') \subseteq V_2(h^s, h')$$

より、

$$a(s_j, v) \in A_1(h_j^s) \text{かつ } a(v, t_j) \in A_1(h_j')$$

であるので、

$$u(a(s_j, v)) \geq \omega(\text{ord}^{-1}(h_j^s)) \text{かつ } u(a(v, t_j)) \geq \omega(\text{ord}^{-1}(h_j'))$$

が成り立つ。よって (ロ) が示された。

(十分性の証明了)

(補題3.7の証明了)

補題3.7が示されたことによって、MCR2γ-blは制約式<3.1>~<3.11>を解く問題に帰着できることが分かった（帰着の操作が入力の多項式時間で出来ることは容易に確かめられる）。

【補題3.8】 制約式<3.1>~<3.11>は多項式時間で解くことができる。□

補題3.8の証明) まず、制約式<3.1>~<3.11>の変数の個数を見積る（但し $a^b := a^b$ とする）。

$x(k, j, \ell, h^s, h')$ は高々 $b \cdot b \cdot 2 \cdot (2^b)^b \cdot (2^b)^b = 2b^2 \cdot 2^{2b^2}$ 個、

$y(h, j, \ell, h^s, h')$ は高々 $2^b \cdot b \cdot 2 \cdot (2^b)^b \cdot (2^b)^b = 2b \cdot 2^{2b^2+b}$ 個、

$z(\eta^s, \eta', h^s, h')$ は高々 $(2^b)^b \cdot (2^b)^b \cdot (2^b)^b \cdot (2^b)^b = 2^{4b^2}$ 個

の変数の数を持つ。 $b \geq 1$ であることを考慮すれば、変数の数は全部で高々 $3 \cdot 2^{4b^2}$ 個。各変数の取りうる値は0以上 $|V|$ 以下の整数である。よって、全ての値の組合せを確かめても、高々 $(|V|+1)^{(3 \cdot 2^{4b^2})}$ 通りの場合を確かめれば良い。

次に、選んだ変数の組合せが、制約式<3.1>～<3.11>を満足するかどうかを確かめるのに必要な計算時間を見積る。

<3.1>は $O((2^b)^b \cdot (2^b)^b \cdot (2^b)^b \cdot (2^b)^b) = O(2^{4b^2})$ 時間、

<3.2>は $O((2^b)^b \cdot (2^b)^b \cdot b \cdot b \cdot 2) = O(b^2 \cdot 2^{2b^2})$ 時間、

<3.3>と<3.4>は共に $O(b \cdot b \cdot (2^b)^b \cdot (2^b)^b) = O(b^2 \cdot (2^b)^b)$ 時間、

<3.5>は $O(2^b \cdot b \cdot 2 \cdot (2^b)^b \cdot (2^b)^b) = O(b \cdot 2^{2b^2+b})$ 時間、

<3.6>と<3.7>は共に $O(b \cdot b \cdot (2^b)^b \cdot (2^b)^b) = O(b^2 \cdot 2^{2b^2})$ 時間、

<3.8>は $O((2^b)^b \cdot (2^b)^b \cdot (2^b)^b \cdot (2^b)^b) = O(2^{4b^2})$ 時間、

<3.9>は $O(2^b \cdot b \cdot 2 \cdot (2^b)^b \cdot (2^b)^b) = O(b \cdot 2^{2b^2+b})$ 時間、

<3.10>と<3.11>は共に $O((2^b)^b \cdot (2^b)^b \cdot 2^b \cdot 2 \cdot (2^b)^b \cdot (2^b)^b) = O(2^{4b^2+b})$ 時間

で確かめることが出来る。よって、<3.1>～<3.11>を満たすかどうかを確認するのは、 $O(2^{4b^2+b})$ 時間でできる。故に全ての組合せについて<3.1>～<3.11>を満たすかどうかを確認するのは、 $O(2^{4b^2+b} \cdot (|V|+1)^{(3 \cdot 2^{4b^2})})$ 時間で出来る。 b は定数であるので、これは $|V|$ の多項式で表される計算時間である。 Q.E.D.

これまでの議論から、次の補題が導かれる。

【補題3.9】 $MCR2\gamma-b$ はクラスPに属する。□

証明) 補題3.7と3.8より、 $MCR2\gamma-b$ は $|V|$ の多項式で表される計算時間 $O(2^{4b^2+b} \cdot (|V|+1)^{(3 \cdot 2^{4b^2})})$ で解くことができる。 Q.E.D.

以上の結果、定理3.6を証明することができる。

定理3.6の証明) 補題3.5と3.6と3.9より明か。 Q.E.D.

なお、このアルゴリズムに従ってMCR2- b を解く計算時間は補題3.3, 3.6, 3.9の証明から、 $O(2^{(4b^2+b)} \cdot |V|^{(2b)} \cdot (|V|+1)^{(3 \cdot 2^{(4b^2)})})$ であり、定数の b に対して $|V|$ の多項式で表される計算時間ではあるが、 b に対して急速に増加する関数であるので、このアルゴリズムの実用性は無い。

定理3.3～3.6の結果を表3.2にまとめる。MCRとMCR2は共に強NP完全であるが、品種数が2以上の定数であるとき、MCRは強NP完全性を保つのに対し、MCR2はクラスPに属する点に本質的な違いがある。

表3.2 MCRとMCR2の計算量の比較

Tab. 3.2. Comparison between MCR and MCR2 by their complexity.

| number of commodities problem | unfixed | fixed | |
|-------------------------------------|------------------------------------|------------------------------------|---|
| | | greater than or equal to 2 | 1 |
| MCR | NP-complete in the strong sense | NP-complete in the strong sense | P |
| MCR2 | NP-complete in the strong sense | P | P |

3. 4. まとめ

本章では通信網への応用に際し、実用上重要な制約の付いた多品種流問題の計算量について考察した。その制約は「経路枝数制約」と「均等配分制約」である。

経路枝数制約については枝数制約値を入力としても多項式時間で解けることを示し、具体的には円環化法を提案した。円環化法は、MCFLを最小コスト多品種流問題(CMCF)に帰着して解く手法である。このため、問題を解く手法として、CMCFの高速アルゴリズムが適用できるという利点がある。ただし、対象とするネットワークに円環化を施すとネットワークの規模が枝数制限値(すなわち l)倍になってしまうという欠点もある。しかし大規模な問題は、指数オーダのアルゴリズムで解くことはできないので、現時点では円環法が、大規模問題を解き得る唯一のアルゴリズムであると言える。

均等配分制約の付いた問題MCRについては、MCRは品種数が定数でも2以上ならば強NP完全であることを示した。そして品種数が1の場合の多項式時間アルゴリズムも示した。次に応用上重要な、経路枝数を2以下に制限した問題MCR2について論じ、一般の場合はやはり強NP完全だが、品種数が任意の定数の場合はクラスPに属することを示した。その多項式アルゴリズムは、多項式の次数が高く実用性は無いが、理論的に多項式時間で解けることを示したことに意味がある。

次に未解決の関連話題について述べる。本章ではグラフは単純グラフとしたが、多重グラフまで拡張しても類似の議論が成立すると予想される。また、MCRの定義の条件(v) (4.3.1参照)で、素(disjoint)を点素の意味で用いたが、枝素と変えても、ほぼ同じことが言える。また、条件(v)を外した場合、経路枝数が2以下の問題ならば変わらないが(なぜなら発着節点が等しい異なる2路は必ず内素であるから)、経路枝数の制限が無い問題は、路の部分集合を求める様な問題になる。単純有向グラフにおいて、異なる路の総数は $|V|!$ 個あり、それらの部分集合の数は $2^{|V|!}$ 個になるため、一般に路の部分集合の総当たりはクラスNPに属するアルゴリズムでは無い。このため本問題(条件(v)が無く、経路枝数の制限も無い問題)が、クラスNPに属するののかも明かでは無い。

本稿では、流量配分則として均等配分則のみを扱ったが、他の流量配分

則についても、品種数が可変の場合はNP困難になるものが見つかっている [ITO 92a]。この検討は、管理者が流れ自由に制御できない、という場合を想定した理論だが、さらに推し進めて、管理者の意思に反して最悪に流れてしまう可能性もあるという前提の問題も考えられ [IRI 94]、この前提に基づく流量配分則の問題の解法及び複雑さも興味あるところである。

第4章

迂回候補群作成問題

4. 1. 通信網制御の問題点とSTR

前章までは、組合せ最適化問題を計算量とアルゴリズムという理論的側面から検討したが、本章では、組合せ最適化手法を用いて具体的問題を解いた例[ITO 92c] [ITO 92d]を示す。本成果はNTTの中継交換網を制御する方式の研究であり、現在この方式は導入され、効果をあげている。

4. 1. 1. ダイナミックルーティングとSTR

中継網での2点(ZC)間の通信経路は、一般に複数存在するが、設計の際にあらかじめ適切な経路に定められている。しかし経済的に網を設計し運用するためには、迂回を利用するのが効率的である。つまり固定された第一選択経路が全塞の場合、適切な経路に迂回することによって網設備が有効利用できるのである。

従来はこの迂回経路も設計時に固定的に与えられていた。しかし、新サービスの導入や複数会社の競合、電話の使用目的の多用化、料金体形の変化等によってトラヒック変動も多様化し、トラヒック予測が困難になりつつある。よって時間帯の変化や網状態の変化に応じて迂回路を変化させた方が効率が良い。この方式をダイナミックルーティング(DR: Dynamic Routing)方式と言う。高性能な計算機を搭載した交換機の導入や情報伝達網の発展に伴ってDRの導入が具体性を帯びてきた80年代前半から、各国でDRの研究開発が盛んに行われている[ASH 81] [ASH 85] [HURL 87] [ASH 90] [CHEM 90] [GIRA 90] [KEY 90] [MASE 90] [REGN 90] [ASH 91] [CHAU 91] [INOUE 91]。

DRには、時間帯毎にあらかじめ決めておいた迂回路に変更していく「時間依存(time-dependent)型」と、何らかの方法で網状態を観測し、変化に対

応して迂回路を変化させていく「状態依存(state-dependent)型」とがある[HURL 87]が、両者ともに一長一短がある。時間依存型は国際網や、米国、カナダの国内網など、網内に時差がある場合に威力を発揮するが、予測困難なトラフィック変動に追随する能力は無い。一方、状態依存型は一見時間依存型の能力を包含する理想的な方式の様に思えるが、実用の観点から見るとやはり難点がある。というのは、状態依存型DRを実現するためのシステム構成は、1か所の網制御センタで全網の情報を管理し、各交換機の指示をあたえる集中制御方式[ASH 85] [CHEM 90] [REGN 90] [CHAU 91]と、数か所の網制御センタごとまたは交換機ごとに個別に得られる情報をもとに制御する分散制御方式[KEY 90] [ASH 91]の2通りがある。分散制御方式を用いた場合、部分的な情報に頼らなければならないので情報が不正確になりがちであるという欠点がある。集中制御方式は全網の情報を利用できるが、反面情報を収集する周期や、転送に要する時間が制御遅延となって性能に影響する。実際、空回線数最大の迂回路を選ぶアルゴリズムである最小負荷経路選択(LLPR: Least Loaded Path Routing)は、制御遅延が、呼の平均保留時間の10分の1（平均保留時間が100秒なら10秒）以内になるように制御しないと、性能が維持できないことが確かめられている[INOUE 89] [MASE 90]。この様に、状態依存型DRを導入する場合、妥当な導入コストで網状態を正確に把握できるシステムを実現することは困難である。

そこで提案されたのが、時間依存と状態依存の両者を合わせた方式、STR(State- and Time dependent Routing) [MASE 90] [INOUE 91]である。STRでは、短時間変動に即応するための分散制御方式による状態依存型DRと、網の定常的な混み具合の情報をを用いた時間依存型DRを合わせて、より正確な迂回路選択を達成するものである。つまり、定常的に空きが多いと予想されるいくつかの迂回路を、あらかじめ限定して交換機毎に与えておき（時間依存）、これらの中から分散制御方式による単純な状態依存型DRによって迂回路を決定する（状態依存）方式である。時間依存部分で、各交換機にあらかじめ与えておく迂回路のことを迂回候補群(routing domain)という。迂回候補群を限定する効果は、(1)適切な迂回路を選択する確率を高めることだけではなく、(2)選択可能な迂回路の数が少なくなるので、迂回処理時の各交換機の処理量やメモリ量を減らす、(3)回線輻輳が直接影響する範囲が限定されるので、網管理が容易になる、という効果も期待できる。

4. 1. 2. 迂回候補群作成の問題点

迂回候補群の作成は、予測される交流呼量データと回線構成データから、迂回呼量を予測し、その迂回呼量を配分する迂回路を限定することである。よって各回線群の余裕を予測し、その余裕の大小と、各迂回呼量の大小を考慮し、呼量配分する必要がある。また適切な呼量配分が求まれば、配分された迂回路の集まりを迂回候補群とすれば良いので、迂回候補群作成問題は呼量配分の問題と考えて良い。呼量配分問題に関してはこれまでも検討されている[ASH 81] [ELSA 88]が、どのアルゴリズムも線形計画問題を繰り返し解く等の複雑な計算を必要とする。

しかし、迂回候補群の利用法としては、数時間から半日単位で変化させることも考えられ[MASE 90]、その場合、多大な処理時間を必要とする複雑なアルゴリズムは望ましくない。迂回候補群作成問題を最適化問題として考えた場合、交換機数を n とすると、発着交換機の組合せの数は $n(n-1)$ あり、夫々の発着交換機の組合せに対して迂回路の数は、2リンク（経路枝数2）の路に限定しても最大 $n-2$ 存在するので、変数の数は $n(n-1)(n-2)$ になりうる。日本の中継網への適用を考えたとき、交換機数は200以上なので、変数の数は $200 \cdot 199 \cdot 198 \div 800$ 万と膨大になり、それを厳密に解くことは困難である。さらに、一つの交換局に複数の交換機が存在するマルチユニット方式[HASI 86a]や、災害等による大規模な設備故障時にも経路の一部確保するための二重帰属[HASI 86a] [HASI 86b]などの網形態を考慮しつつ最適解を求めることはほぼ不可能である。

STRでは、時間依存型DR方式における固定迂回とは異なり、その中で状態依存型DRを行うので、迂回候補群の最適解からのずれを補うことができる可能性がある。本章では、日本の中継網にも導入可能な、STRの迂回候補群作成アルゴリズムとして、高速な発見的算法である積上げ法を提案し、性能を評価する。4. 2で積上げ法のアルゴリズムとそれを用いたDR方式の効果を示す。4. 3と4. 4では積上げ法の妥当性を様々な角度から検証する。4. 3で、迂回候補群作成問題を非線形の数理計画問題に定式化し、線形計画法の繰り返しで解くことによって得られた迂回候補群と、積上げ法による迂回候補群とを比較する。4. 4で、積上げ法で作成された迂回候補群に含まれる迂回路は、理想的な状態依存型DR方式でも高い確率で選択されることを、シミュレーションを用いて確認する。

4. 2. 迂回候補群作成問題と積上げ法

4. 2. 1. STRの概要

既に述べた様に、予測呼量データに基づいて迂回候補群を与え、そのうえで状態依存型DRを行う方式をSTRと呼んでいる[MASE 90] [INOUE 91]。以下にSTR方式の概要を記す。

数時間～半日単位に区切った時間帯毎に呼量を予測し、それに基づいて網管理センタで迂回候補群を作成し、各交換機に通知する。各交換機は与えられた迂回候補群の中から、各交換機が独立に迂回路を変更していく。この迂回路変更のアルゴリズムとしては、自律迂回路選択方式[INOUE 89] [KEY 90] [MASE 90] [INOUE 91]を用いている。この方式は、

「迂回候補群より迂回路を1つ登録しておき、第1選択路が全塞の時、まず、登録された迂回路を用いて迂回の可能性を見る。その迂回路で迂回できなかった時、その呼は呼損となる（1段迂回）。この場合、その迂回路を登録からはずし、迂回候補群の中から、次に設定されている路を選び、新たに迂回路として登録する。」

という方法であり、学習ルーチングの最も単純なアルゴリズムでありながら高い性能が得られることが解っている。

4. 2. 2. 迂回候補群作成問題

通信網を有向グラフ $G=(V,A)$ で現す。 G 上の節点 $v \in V$ は交換局に、枝 $a(v,w) \in A$ は局間回線群に対応する。各枝には回線数が付されている。与えられた G と各節点間（発着局間）呼量に対して望ましい迂回候補群を求める問題を迂回候補群作成問題と呼ぶ。この問題は、前提とする網構成及びルーチング方式により、定式化が異なる。本稿では以下を前提とする。

[前提]

- (1) 対象は即時系の回線交換網。
- (2) 任意の節点が呼の発着信及び中継の処理をする非階層網で、相互迂回がある。
- (3) 各節点間に第1選択路（基本路と呼ぶ）が与えられている。各呼は基

本路をまず選択し、基本路に空きが無い場合、他の路に迂回する。

(4)基本路を選択しようとする呼（基本呼と呼ぶ）を、他の基本路から迂回してきた呼（迂回呼と呼ぶ）よりも優先して接続するために、回線留保（付録3参照）の措置がとられる。

(5)各交換機内での呼損は微小であるので、これを無視する。

(6)迂回路は枝毎に与えられ、各迂回路は2つの枝からなる路（迂回可能路と呼ぶ）に限る。

4. 2. 3. 積上げ法

積上げ法(cumulative method)は、迂回候補群を求めるための発見的算法で、予測される節点間呼量と各枝の回線数、留保回線数を入力データとする。

その基本的な考え方は、各枝からの溢れ呼量を、各迂回可能路に、それらの受け入れ可能な迂回呼量に応じて割り当てていく、というものであり、次の2つの処理からなる。

(i)余裕呼量、要求呼量の計算

呼量と回線数より、枝毎に次の2つの値を計算する。

要求呼量：基本呼のみが各枝に加わると仮定したときの各枝からの溢れ呼量。

余裕呼量：基本呼の基準品質（呼損率：入力値）を満足する範囲で各枝が受け入れることのできる迂回呼量。

(ii)呼量割当

以下の操作を繰り返す。

最も要求呼量の大きい枝を選び、その枝の迂回可能路の中で最も余裕呼量の大きい路（迂回可能路の余裕呼量は構成する枝の余裕呼量の小さい方の値で定義する）に、要求呼量の一部を割り当てる。

積上げ法はこの様に、要求呼量を、余裕呼量の大きい枝から構成される迂回可能路に割り当てていくという、貪欲算法に基づいて、呼量配分の問題を解くアルゴリズムである。積上げ法は、複雑かつ大規模な網への適用を目的としているため、処理の簡略化、高速化を重視している。その実現

のために以下の点を工夫している。

- ・ 余裕呼量、要求呼量は初期値算出時のみ確率性を考慮し、アーラン1式を用いる。
- ・ 呼量割当では、流量割当問題として呼量を扱い、呼の確率性は考慮しない。
- ・ 流量割当は近似算法によって解く。

以下に積上げ法のアルゴリズムCUMULATEの詳細を記す。なお説明を簡単にするために以下、本稿を通じて、完全網状網を前提とし、基本路は直通の枝とした。

procedure CUMULATE

comment

入力

$trf(v,w)$: 節点 v 発、節点 w 着の呼量

$n(v,w)$: 枝 $a(v,w)$ の回線数

$K(v,w)$: 枝 $a(v,w)$ に与えたい迂回候補数

b_0 : 呼損率基準値 (通常 $b_0=0.01$)

出力

$R(v,w)$: 枝 $a(v,w)$ の迂回候補群

その他

$d(v,w)$: 枝 $a(v,w)$ の要求呼量

$cap(v,w)$: 枝 $a(v,w)$ の余裕呼量

$unit(v,w)$: 枝 $a(v,w)$ の要求呼量を割り当てる単位

$F(v,w)$: 枝 $a(v,w)$ の迂回可能路集合

Y : 迂回候補群の設定未終了の枝の集合

$B(n,trf)$: 回線数 n 、呼量 trf の時の呼損率関数 (1. 5 参照) ;

begin

1 begin

2 for each $a(v,w) \in A$ do

3 $d(v,w) := trf(v,w) \cdot B(n(v,w), trf(v,w))$

4 $b_0 = B(n(v,w), trf(v,w) + cap)$ を満たす cap を求め、 $cap(v,w) := cap$

5 $unit(v,w) := d(v,w) / K(v,w)$

```

6       $R(v,w) := \phi$ 
7       $F(v,w) := \{u \in V \mid a(v,u) \in A, a(u,w) \in A\}$ 
8       $Y := \{a(v,w) \in A\}$ 
      end;
9  begin
10   begin
11       $d(x,y) = \max_{a(v,w) \in Y} d(v,w)$  となる、 $a(x,y) \in Y$  を選ぶ
12      if  $|R(x,y)| < K(x,y)$  then
13         if  $d(x,y) > 0$  then
14             $\min(\text{cap}(x,z), \text{cap}(z,y)) = \max_{u \in R(x,y)} \min(\text{cap}(x,u), \text{cap}(u,y))$ 
               となる  $z \in F(x,y)$  を選ぶ
15         else
16             $\min(\text{cap}(x,z), \text{cap}(z,y)) = \max_{u \in R(x,y) - R(x,y)} \min(\text{cap}(x,u), \text{cap}(u,y))$ 
               となる  $z \in F(x,y) - R(x,y)$  を選ぶ
17         fi;
18      else
19          $\min(\text{cap}(x,z), \text{cap}(z,y)) = \max_{u \in R(x,y)} \min(\text{cap}(x,u), \text{cap}(u,y))$ 
               となる  $z \in R(x,y)$  を選ぶ
20         fi;
21      end;
22   begin
23       $R(x,y) := R(x,y) \cup \{z\}$ 
24       $A := \min(\text{unit}(x,y), d(x,y))$ 
25       $d(x,y) := d(x,y) - A$ 
26       $\text{cap}(x,z) := \text{cap}(x,z) - A$ 
27       $\text{cap}(z,y) := \text{cap}(z,y) - A$ 
28   end;
29   begin
30      if  $|R(x,y)| = K(x,y)$  and  $d(x,y) = 0$  then
31          $Y := Y - \{a(x,y)\}$ 
32      fi;
33   end;
34   if  $Y \neq \phi$  then goto line 10;

```

end;
end.

本アルゴリズムの意味を簡単に説明する。

第1～8行は余裕呼量、要求呼量の初期値の計算である。第9行以降では余裕呼量に要求呼量を割り当てていく。第10～18行で、差し当たって呼量割当を行なう枝と迂回路の選択をする。第19～24行では、選ばれた枝の要求呼量の一部を選ばれた迂回路に割り当てた結果として、余裕呼量と要求呼量を更新する。第25～28行では、終了判定を行ない、終了条件（迂回候補群が要求本数 $K(x,y)$ だけ得られ、かつ全ての要求呼量がゼロ）を満たしていなければ、繰り返し第10行以降の操作を行なっていく。

次に計算量を見積る。

第10行目以降の繰り返しの回数は高々

$$\sum_{v,w \in V} (d(v,w)/unit(v,w) + K(v,w)) = 2 \sum_{v,w \in V} K(v,w) \quad (\text{第5行より}) \\ \leq |A|K \quad (\text{但し } K = \max_{v,w \in V} K(v,w))$$

回である。一回の繰り返しで必要な計算量は $O(|V|)$ である。よって、初期値の計算（第1～8行目）を除けば計算量は $O(|V||A|K)$ となる。初期値の計算は、枝数回の呼損率関数の計算を必要とする。よって回線数 n , 呼量 trf の回線の呼損率計算に必要な計算量を $f(n, trf)$ とおくと、初期値の計算は $O(|A|f(N, T))$ 時間必要である、但し $N = \max_{x(v,w) \in A} n(v,w)$, $T = \max_{x(v,w) \in A} trf(v,w)$ とする。よって積上げ法の計算量は

$$O(|V||A|K) + O(|A|f(N, T))$$

である。ここで、 $f(N, T)$ の大きさが問題であるが、呼損率関数の計算量は厳密に行なうと、回線数に比例する計算時間が必要であり、公衆中継網の様な大量の呼をさばく網においては、無視できない計算時間を必要とする。但し、高速近似算法も提案されており、一つの呼損率関数の計算が $O(1)$ で出来る近似算法も提案されている[JAGE 74]。本研究の計算実験では、厳密な呼損率計算を採用したが、積上げ法では、枝数回の呼損率計算で十分であり、迂回候補群作成を行なう為には最小限の計算回数であると考えられる。よって、初期値計算の部分の計算量 $O(|A|f(N, T))$ は必要最小限であり、後半の計算量 $O(|V||A|K)$ も、迂回候補群を $|A|K$ 本選ばなければならないことを考えると、1本当たり $O(|V|)$ 時間であり、計算量は十分小さいことが分かる。

4. 2. 4. 迂回候補群の効果

積上げ法により迂回候補群を作成することによって、DR運用時の網の呼損率を低減することができる。それをシミュレーション[INOUE 89]によって確認する。

この評価では、網内の各基本路に与えられる迂回候補数を等しくし、その数を変化させた時の各評価値の変化の様子を示した。迂回候補群作成法としては、比較のために積上げ法の他に、回線数や予測呼量と無関係に、乱数で無作為に迂回候補群を与える方法も用いた。状態依存型DR方式は自律迂回路選択（4. 2. 1 参照）を用いた。

シミュレーションの結果を図4.1、4.2に示す。評価尺度は平均呼損率、最悪品質対地間の呼損率（以下最悪呼損率と呼ぶ）、呼損率限界値（GOS; Grade Of Service; 今回は3%に設定した）を超過した対地数を用いた。図4.1は小規模なモデルA、図4.2は大規模なモデルB（詳しくは表4.1参照）を用いた場合の評価結果である。

モデルA、Bとも、無作為に迂回候補群を作成すると、迂回候補数が少ないほど評価値は劣化した。しかし積上げ法を用いた場合は、迂回候補群をある程度絞ったところ（10局網のモデルAでは2~3、36局網のモデルBでは2~8）で各評価値は最も良い値を示し、迂回候補数が多すぎても少なすぎても各評価値は劣化した。この理由は、以下の2つが考えられる。

(1) 迂回候補数が多すぎると定常的に混んでいる迂回可能路も迂回候補群に選ばれてしまう。

(2) 迂回候補数が少なすぎると十分に迂回呼を疎通するだけの迂回候補群が用意されない。

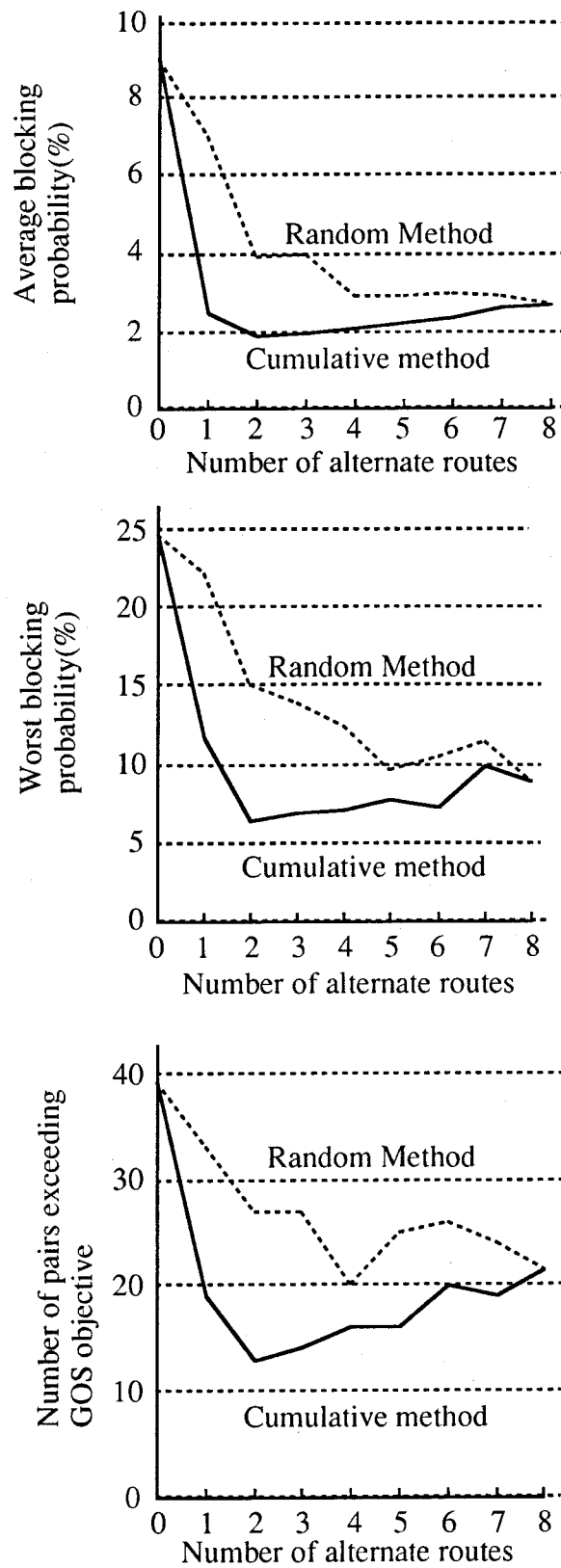


図4.1 10局網における迂回候補群の効果

Fig. 4.1 Effect of a routing domain on a 10-node network.

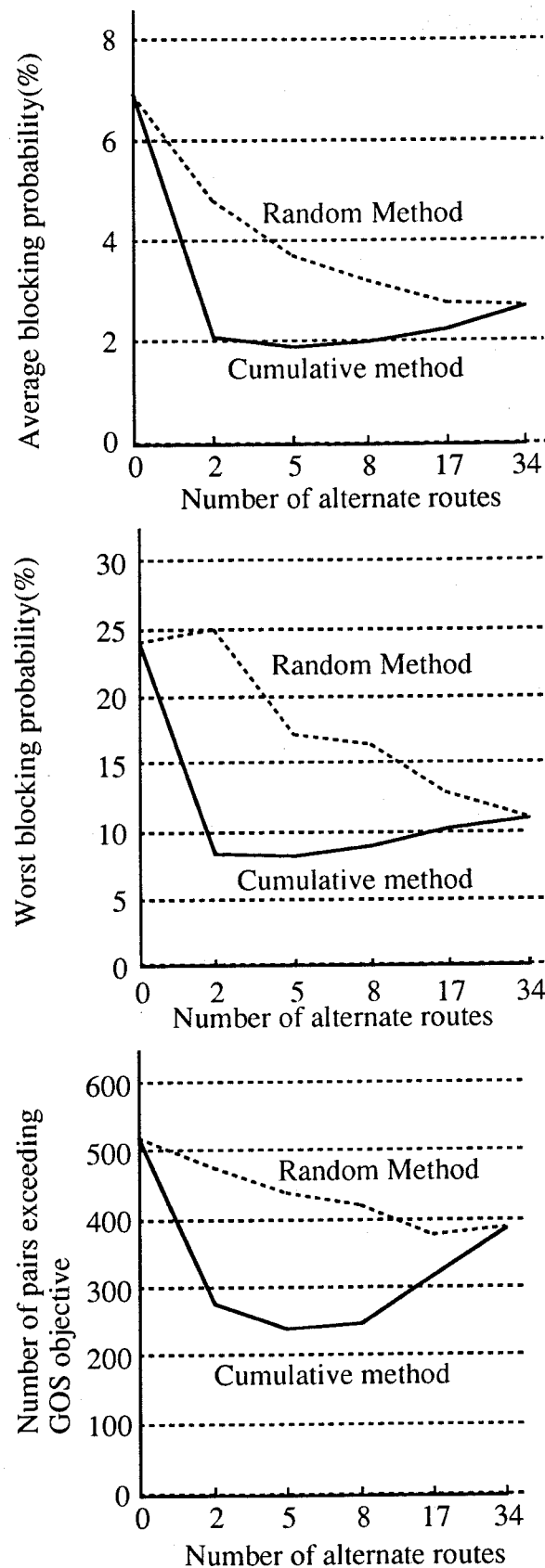


図4.2 36局網における迂回候補群の効果
 Fig. 4.2 Effect of a routing domain on a 36-node network.

4. 2. 5. 迂回候補数の算出法

積上げ法では枝毎の迂回候補数 $K(v,w)$ を入力値とするため、適当な迂回候補数の決定法が必要である。

前節で示したように与えられた網と交流呼量に対して、最適な迂回候補数が存在する。

迂回候補数の上限は、4. 2. 4に記した(1)の理由で、迂回呼（非優先呼）の呼損率の低い枝より成る迂回可能路で構成されなければならない。迂回呼の呼損率は、基本呼の呼量が設計値を越える前後で急激に悪化するので、迂回候補群に使用する枝は、設計品質を満足している必要がある。また、迂回候補数の下限については、4. 2. 4に記した(2)の理由で溢れ呼を処理できる本数が最低必要となる。この上下限は具体的には式(4.1)で表される。

$$q(v,w) \leq K(v,w) \leq p(v,w) \quad (4.1)$$

但し、 $q(v,w)$ は、 $d(v,w)$ 、 $cap(v,w)$ を各々枝 $a(v,w)$ の要求呼量、余裕呼量の初期値としたとき、

$$d(v,w) \leq \sum_{u \in F(v,w)} \min(cap(v,u), cap(u,w))$$

を満足する迂回可能路の部分集合 $F'(v,w) \subseteq F(v,w)$ で要素数最小のものの要素数であり、積上げ法のアルゴリズムで求めた $cap(v,w)$ 、 $d(v,w)$ の初期値を用いて簡単に計算できる。また、 $p(v,w)$ は枝 $a(v,w)$ の迂回可能路の内、設計品質を満たしているものの数を表し、積上げ法のアルゴリズムにおいて、 $cap(v,w)$ の初期値を計算した時、 $cap(v,w) > 0$ となる枝 $a(v,w) \in E$ の数を数えることによって得られる。

最適な迂回候補数は式(4.1)の範囲に存在する。しかしその最適値を厳密に求めなくても、式(4.1)に示す範囲の中間に指定すれば実行上は十分である。その理由は、ここで前提とするDR方式では、各発着信局間に対して基準品質を満足する基本路を第一に選択するため、迂回路は溢れ呼の吸収に用いられるので、最適な迂回候補数からのずれが小さければ、接続品質の劣化はほとんど無視できるからである。

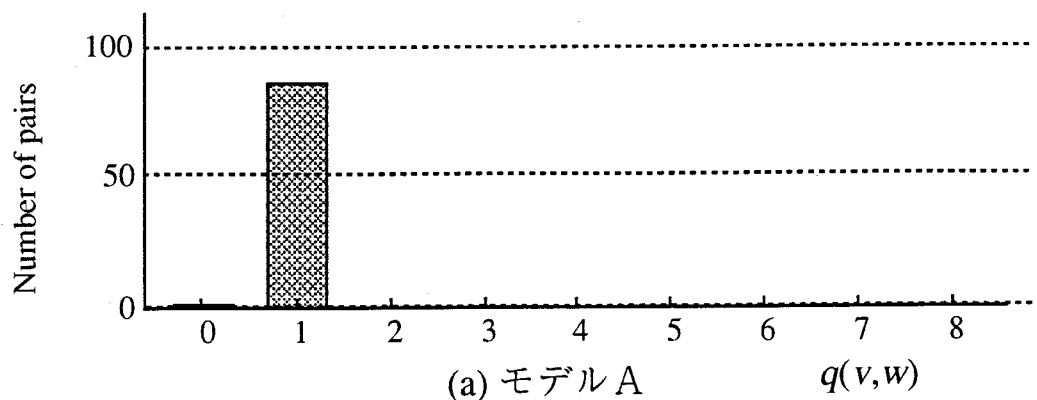
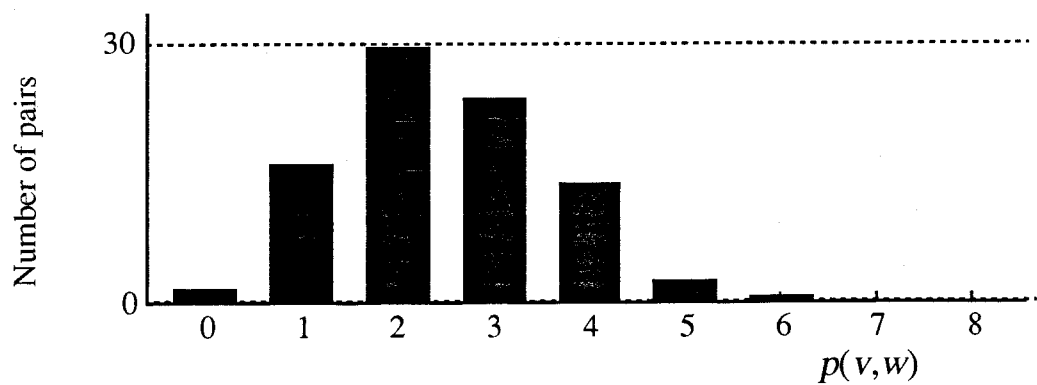
モデルA、Bにおける $q(v,w)$ 、 $p(v,w)$ の値の分布は大体、

モデルAでは $q(v,w)=1$ 、 $p(v,w)=1 \sim 4$

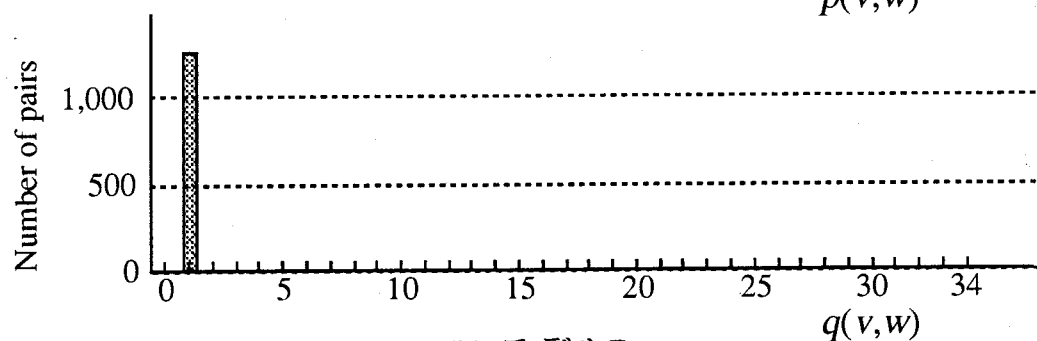
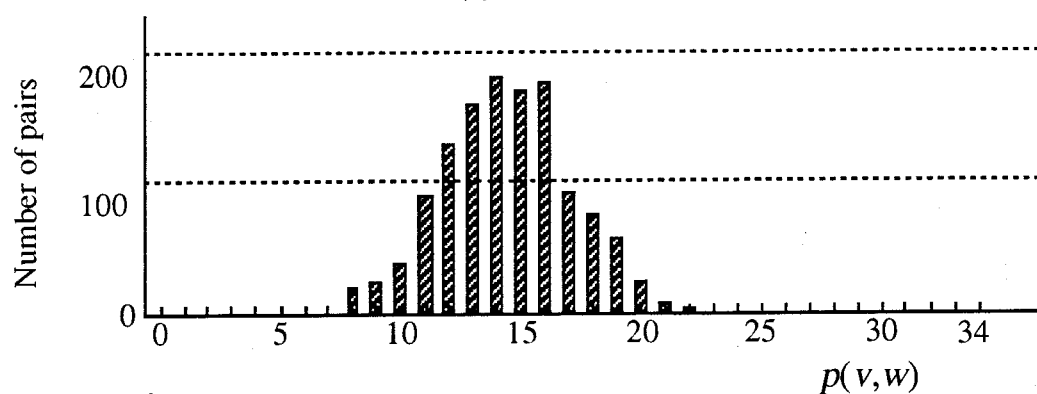
モデルBでは $q(v,w)=1$ 、 $p(v,w)=8 \sim 20$

(4.2)

となった（図4.3参照）。



(a) モデル A
(a) Model A



(b) モデル B
(b) Model B

図4.3 $p(v,w)$ と $q(v,w)$ の分布の例

Fig.4.3 Examples of distributions of $p(v,w)$ and $q(v,w)$

本章では完全グラフを対象とするので、簡単のため $K(v,w)$ は枝によらず一定値 K とした。シミュレーションの結果では、

モデルAでは $2 \leq K \leq 3$ 、

モデルBでは $2 \leq K \leq 8$

で良い性能を示した（図4.1、4.2参照）。これは式(4.1)、(4.2)にほぼ一致する。但し $K=1$ の時に比較的性能が良くないが、これは $K=1$ では呼量配分の柔軟性が少ないため、配分が上手く行われないことが原因であると考えられる。現実には $p(v,w)$ の平均値と、 $q(v,w)$ の平均値の中間に設定すれば問題は無い。

4. 3. 線形計画法に基づく迂回候補群作成法との比較

4. 2. 4に記した結果は、積上げ法を用いて迂回候補群を作成することが効果的である事を示している。しかし、積上げ法のアルゴリズムは前述のように、高速性を追求した発見的算法なので、理論的な厳密さは無い。よって、迂回候補群作成アルゴリズムをさらに高度なものにすればDRの性能を上げられるかもしれない。しかしそのアルゴリズムが複雑なもので、長時間の計算時間が必要なものならば、実現性に問題がある。4. 3と4. 4では、これらのことを考慮して、積上げ法の妥当性を検証する。

本節では、迂回候補群作成法を積上げ法の様な発見的算法ではなく、数理計画問題に定式化し、それを解くことによって得られる迂回候補群と、積上げ法によって得られる迂回候補群を比較する。

4. 3. 1. 迂回候補群作成問題の定式化

迂回候補群作成問題は、各枝の溢れ呼を各迂回可能路に割り当てる呼量割当問題と考えられる。目的関数には、網全体の呼損率を最小にすることを取り、問題を解いて、迂回呼が割り当てられた迂回可能路を迂回候補群であると判断する。呼量割当問題は、呼損率関数が凸、迂回候補数が各枝1本ずつであると仮定してもNP困難問題になることが知られている[ELSA 88]。しかし準最適解を求める方法は種々検討されている[ASH 81][ELSA 88]。

迂回候補群作成問題では回線数と呼量は固定であるため、次の様に定式化される。なお、以下では溢れ呼のポワソン性を仮定した。この仮定はこの類の問題ではよく用いられる。

以下にまず使用する記号を定義したのち、問題を定式化する。

【定義】

入力

$G=(V,A)$: ネットワークを表す有向グラフ

$trf(v,w)$: 節点 $v \in V$ 発、節点 $w \in V$ 着の呼量

$n(v,w)$: 枝 $a(v,w) \in A$ の回線数

$m(v,w)$: 枝 $a(v,w) \in A$ の留保回線数

出力

$x(v,u,w)$: 枝 $a(v,w) \in A$ の溢れ呼のうち迂回可能路 $\langle v,u,w \rangle, u \in V, u \neq v, w$ に割り当てられる割合。但し $a(v,w) \notin A$ または $u=v$ または $u=w$ の時 $x(v,u,w)=0$ とする。

その他

$trf_1(v,w)$: 枝 $a(v,w) \in A$ にかかる基本呼量

$trf_2(v,w)$: 枝 $a(v,w) \in A$ にかかる迂回呼量

$b_1(v,w)$: 枝 $a(v,w) \in A$ の基本呼の呼損率

$b_2(v,w)$: 枝 $a(v,w) \in A$ の迂回呼の呼損率

$o(v,w)$: 枝 $a(v,w) \in A$ からの溢れ呼量

$trf_c(v,w)$: 節点 $v \in V$ 発、節点 $w \in V$ 着の呼で完了した呼量

$B_1(n,m,trf_1,trf_2)$: 回線数 n 、留保回線数 m 、優先呼（基本呼）量 trf_1 、非優先呼（迂回呼）量 trf_2 の時の優先呼の呼損率を与える関数（付録3 参照）

$B_2(n,m,trf_1,trf_2)$: 回線数 n 、留保回線数 m 、優先呼（基本呼）量 trf_1 、非優先呼（迂回呼）量 trf_2 の時の非優先呼の呼損率を与える関数（付録3 参照）

問題ALT

[目的関数] $\sum_{a(v,w) \in A} trf_c(v,w) \rightarrow \max$ (4.3)

[制約条件]

$$trf_c(v,w) = trf_1(v,w)(1-b_1(v,w)) + \sum_{u \in V} x(v,u,w)o(v,w)(1-b_2(v,u))(1-b_2(u,w)), a(v,w) \in A \quad (4.4)$$

$$o(v,w) = trf_1(v,w)b_1(v,w), a(v,w) \in A \quad (4.5)$$

$$trf_2(v,w) = \sum_{u \in V} x(v,w,u)o(v,u)(1-b_2(w,u)) + \sum_{u \in V} x(u,v,w)o(u,w)(1-b_2(u,v)) \quad (4.6)$$

$$b_1(v,w) = B_1(n(v,w),m(v,w),trf_1(v,w),trf_2(v,w)), a(v,w) \in A \quad (4.7)$$

$$b_2(v,w) = B_2(n(v,w),m(v,w),trf_1(v,w),trf_2(v,w)), a(v,w) \in A \quad (4.8)$$

$$\sum_{u \in V} x(v,u,w) = 1, a(v,w) \in A \quad (4.9)$$

$$x(v,u,w) \geq 0, a(v,w) \in A, u \in V \quad (4.10)$$

□

各式の意味を簡単に説明する。

目的関数(4.3)は網全体で完了した呼量を最大にすることを意味する。

制約条件(4.4)は $v \in V$ 発、 $w \in V$ 着の完了呼は基本路で完了した呼と迂回路で完了した呼から成ることを表す。

制約条件(4.5)は枝 $a(v,w)$ からの迂回呼量は、枝にかかる基本呼量とその呼損率の積で求められることを表す。

制約条件(4.6)は枝 $a(v,w)$ にかかる迂回呼量は、枝 $a(v,w)$ を、迂回路の第一番目の枝として使用する迂回呼量と、迂回路の第二番目の枝として使用する迂回呼量の和になることを表す。

制約条件(4.7)と(4.8)は枝 $a(v,w)$ の基本呼と迂回呼の呼損率を表す式である。

制約条件(4.9)と(4.10)は $x(v,u,w)$ が迂回路に割り当てられる率を意味することから、当然満たすべき性質である。

問題ALTの式の中で、 $trf_1(v,w)$ は定数 ($=trf(v,w)$) であり、 $trf_2(v,w)$, $trf_c(v,w)$, $o(v,w)$, $b_1(v,w)$, $b_2(v,w)$, $x(v,u,w)$ は変数である。積上げ法はこの問題を解く発見的アルゴリズムと考えることができる。

呼損率を表す関数 B_1 , B_2 は付録3に示す様に非線形関数で、しかも凸ではないので、最適解を得る事は難しい。こういった問題を解く場合は、非線形関数をその時点の解の周辺で線形近似することで、線形計画問題LPを作りその解を算出し、その解に基づいて再びLPに近似することを繰り返して、収束解を求める方法が有効である[ASH 81]。本稿でもその手法を用いる。

反復毎のLPにおいては、 $trf_1(v,w)$ に加えて $b_1(v,w)$, $b_2(v,w)$, $o(v,w)$ も定数と考え $trf_2(v,w)$, $trf_c(v,w)$, $x(v,u,w)$ を変数と考える。しかしこれだけの仮定で問題を解くと、各 $x(v,u,w)$ は $(1-b_2(v,u)) \times (1-b_2(u,w))$ が最大の迂回可能路に集中してしまい、LPを反復しても解が振動するだけで収束しない。LPの係数である $b_1(v,w)$, $b_2(v,w)$, $o(v,w)$ を計算するときに、その時点で得られている $trf_2(v,w)$ を入力として用いるので、LPの出力として得られる $trf_2(v,w)$ は入力と大きく異なることはできない。また逆に、 $trf_2(v,w)$ が入力との変化が少なければ、 $b_1(v,w)$, $b_2(v,w)$, $o(v,w)$ も入力との変化が少ない事が判る。そこで各反復毎に $trf_2(v,w)$ に適当な上限 ($trf_U(v,w)$) と下限 ($trf_L(v,w)$) を付ける。反復が進む毎にその上下限を絞っていけば、解は収束する。このアルゴリズムを「繰り返し法(iterative method)」と名付けることにする。繰り返し法

の詳しい記述を付録4に掲げる。

4. 3. 2. 評価結果

積上げ法によって作成した迂回候補群と、繰り返し法によって作成した迂回候補群を、自律迂回路選択（4. 2. 1 参照）を用いたDRのシミュレーションによって、呼損率、GOS超過対地数を評価尺度として比較した。また無作為に作成された迂回候補群とも比較した。評価にはモデルA（表4.1参照）を用いた。その結果（図4.4参照）、積上げ法は繰り返し法とかなり近い性能を示している。

繰り返し法では解は最適解が得られる保証は無いが、しかし問題ALTの目的関数である平均呼損率は、繰り返し法の方が積上げ法より良くなっている。このことは繰り返し法のアルゴリズムの近似性能の高さを示している。しかし、目的関数ではない最悪呼損率は、逆に繰り返し法の方が悪くなっている。この理由は、繰り返し法によって最適解が得られているわけでは無いことの他に、数理計画問題に定式化して解いた場合は、その目的関数（この場合の平均呼損率）は確かに改善されるが、目的関数以外（この場合の最悪呼損率）については、考慮されないことも原因として考えられる。この様に、繰り返し法で得られた迂回候補群の方が、積上げ法で得られた迂回候補群より劣る場合もある。

数理計画問題で複数の目的関数を持つ問題は多目的計画問題となり、その解法はさらに複雑になる。それに対し、積上げ法の計算量は4. 2. 3で述べたように大変少なく、実用的である。

迂回候補群作成に要した計算時間は、VAX-6330計算機を用いて、積上げ法は10秒以内であるのに対し、繰り返し法は2日～3日かかっている。これは10局網であるモデルAの結果であり、NTTの中継網（ZC数36）を意識したモデルBを繰り返し法で解くことはほぼ不可能である。それに対して、積上げ法ではモデルBでも、5分以内で計算することができる。

以上のことから、迂回候補群作成法を数理計画的に高度な解法を適用しても、大量の計算時間を必要とするにもかかわらず、性能があまり改善されないことが判る。

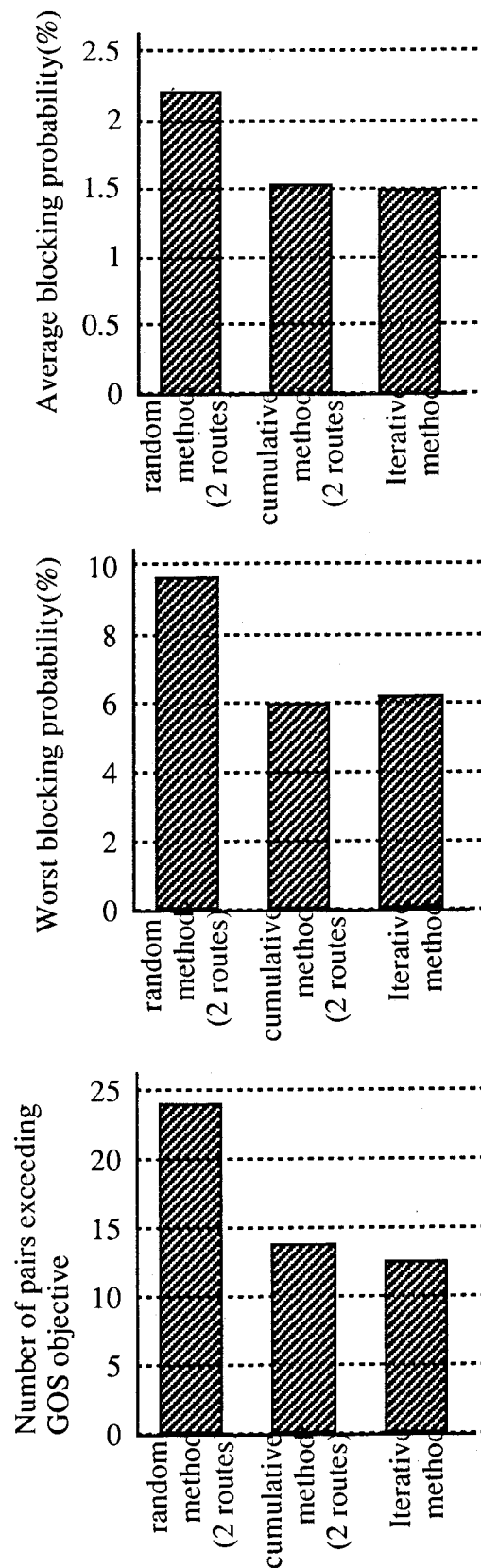


図4.4 積上げ法と繰り返し法の比較

Fig. 4.4 Comparison of cumulative method and iterative method.

4. 4. 最小負荷経路選択法との比較

4. 4. 1. 状態依存型DR法と迂回候補群

理想的な状態依存型DR方式の1つと考えられる最小負荷経路選択法(LLPR)は、空き回線数の最も多い迂回可能路を迂回路に選ぶアルゴリズムであり、TSMR(Trunk Status Map Routing)方式[ASH 85]がこのアルゴリズムを利用している。このアルゴリズムは、空き回線数データが瞬時に得られるという実現不可能な理想状態を仮定したならば、非常に高い性能を示す。LLPRは呼を接続しながら呼の配分を行っているので、呼量が定常状態にあるならば、迂回候補群作成問題を解いているのと等しい。よって、LLPRでの選択率が高い迂回可能路が、迂回路としての性能が高いと考えて良い。そこで、迂回候補群を限定せずにLLPRを適用して各迂回可能路の選択率を求め、迂回候補群に選ばれている迂回可能路の選択率が高いかどうかで、その迂回候補群の適切さを判定することができる。本節では、これによって積上げ法の妥当性を確かめる。

4. 4. 2. 評価結果

本評価はモデルA(表4.1参照)を用いた。迂回候補群を絞り込まずにLLPRを適用したシミュレーションを行い、各迂回可能路で完了した呼数を求めた。迂回候補数を2として積上げ法で作成した迂回候補群に含まれる迂回可能路で完了している呼数(C_1)の、全完了迂回呼数(T)に占める割合は

$$C_1/T=56.6\%$$

であった。それに対して、完了呼数の多い迂回可能路の上位2つで完了している呼数(C_2)の全完了迂回呼数(T)に占める割合は

$$C_2/T=59.0\%$$

であった。よって両者の比は

$$(C_1/T)/(C_2/T)=C_1/C_2=95.9\%$$

となり、積上げ法とLLPRとの、迂回路としての適切さの判断がほぼ一致していることが確かめられた。迂回候補数を1~5に変化させたものについても同様な比較を行った(表4.2参照)が、ほぼ同じ結果が得られている。

表4.1 シミュレーションに用いたモデル

Tab. 4.1 Models used by the simulations.

| | モデルA | モデルB |
|---------------|-----------------------|------------------------|
| 節点数 | 1 0 | 3 6 |
| 回線数 (設計呼量) | 3 0 (2 0 erl) | 2 4 0 (2 1 8 erl) |
| 負荷呼量 の分布 | 20erl±80%の 範囲で一様分布 | 218erl±40%の 範囲で一様分布 |

表4.2 積上げ法とLLPRの上位K本で
呼が完了した確率の比較

Tab. 4.2 Comparison between cumulative method and
LLPR by traffic ratio completed by K -th highest
used routs.

| 迂回候補数 (K) | 積上げ法 (C_1/T) | LLPR (C_2/T) | C_1/C_2 |
|------------------|---------------------|---------------------|-----------|
| 1 | 37.52% | 39.29% | 95.50% |
| 2 | 56.59% | 59.01% | 95.90% |
| 3 | 70.21% | 72.27% | 97.15% |
| 4 | 79.45% | 81.48% | 97.51% |
| 5 | 85.93% | 88.39% | 97.22% |

4. 5. まとめ

状態依存型DR方式を実行する場合、予測呼量に従ってあらかじめ迂回候補群を限定しておくことにより、性能を向上させることができる。その迂回候補群を作成するアルゴリズムとして、「積上げ法」を提案し、以下の点を明らかにした。

積上げ法は、単純で高速であるうえ、性能も線形計画問題への近似を繰り返して解くという「繰り返し法」の様な、複雑なアルゴリズムとほぼ同等であることを示した。繰り返し法の様に数理計画問題に定式化して解く方法は、目的関数は確かに改善されるが、目的関数以外の要因は無視されるため、逆に積上げ法の様な発見的算法より劣ることがあるということも解った。また、積上げ法で作成された迂回候補群に含まれる迂回路は、理想状態を仮定したアルゴリズムであるLLPRでも高い確率で選ばれることが解った。

DR方式を導入する場合には、管理の都合上、迂回候補数をオペレータの判断で変更できる事が望ましい。積上げ法は迂回候補数を入力として指定することができる点でも優れている。逆に性能が最良になるように迂回候補数を指定する必要がある場合、積上げ法では何らかの方法でその数を決定しなければならないが、その場合の適切な迂回候補数の範囲を示した。その範囲の中で最適な数の特定方法は今後の検討を要するが、実用上はその中間あたりに指定しておけば問題無い。

以上のことから積上げ法が、迂回候補群作成の実用アルゴリズムとして非常に有効であることが解る。実現できる状態依存型ルーティングには、状態観測の遅延や、部分情報に頼ることによる情報誤りがあるので、積上げ法の様なアルゴリズムで迂回候補群を絞り込んでおいて、定常的な負荷状態を反映させることは効果的である。

本稿では、迂回候補群作成時の予測呼量とシミュレーション時に加える呼量を等しくしたが、現実のシステムの場合、実際に加わる呼量が、予測した呼量と同じとは限らない。しかしその後の検討で、積上げ法は、ある程度の予測誤差に対して、性能を維持することが判っている[ITO 91] [ITO 92d] [ITO 92e]。

STR方式は1992年よりNTTの中継網に段階的に導入され、効果を上げ、徐々に適用範囲が拡大されている。

第5章

結論

本論文の成果の概要を以下にまとめる。

第2章では、グラフに節点の部分集合（領域）族を追加した「領域グラフ」という概念を導入し、領域グラフ上での、節点と領域間の連結度（NA連結度）について諸定理を明かにし、NA連結に関する種々の最適化問題の複雑さを明かにし、近似アルゴリズムを与えた。グラフ上の領域は、通信網においては、複数の交換機が集まって構成される論理的交換領域のモデル化である。この結果、通信網をより正確にモデル化できるようになり、通信網の強度をより正確に分析できるようになった。領域グラフとNA連結度は、従来のグラフと連結度の拡張であり、汎用性がある。

第3章では、経路枝数および流量配分則に制限のある多品種流問題の複雑さを明かにし、多項式時間アルゴリズムを与えた。この結果、クラスPに属する問題もいくつか明かになった。通信網では、多数の交換機を経由した接続は、品質の劣化や網輻輳の原因となるので、経路枝数制限は非常に現実的な制約である。また、流量配分則は、選ばれた各経路に均等に配分するという均等配分則についてのみ検討した。この制約の通信網での意味を考えると、各交換機での経路選択が、無作為あるいは順番に行なうものが均等配分則に従うことになり、これも使用頻度の高い規則である。

第4章では、ダイナミックルーティングのための迂回候補群作成問題に対し、発見的な高速アルゴリズムを適用して効果をあげた。その際、数値計画法に基づく解法等とも比較し、提案した算法の性能の高さを確認した。ここで提案したアルゴリズムはNTTの中継網の制御法に既に導入されているSTR方式の一部として採用されている。

本研究で扱った問題や概念については、これまで理論的な検討は殆どなされていなかったが、通信網管理上大変重要なものである。その上、これらの問題や概念は汎用性があり、様々な適用があると思われる。

第2章で提案した領域グラフは、複数の部分網の集まりで構成されている網一般がこの形に定式化される。例えばNTTの中継網でも、東京、大阪

等全国で10の網管理センタがあり、夫々が関東圏、近畿圏等を部分網と考えて管理している。また、自動車電話網、パケット通信網等のサービス別の網をそれぞれ部分網と考えることもできる。国際網は各国毎に部分網があると考えられる。インターネットには多くのサブドメインが存在する。これらも部分網であり、領域グラフの概念でモデル化される。

第3章で扱った多品種流問題は、通信網に限らず交通流等物流の問題も定式化されるが、この場合も長大な経路で輸送することは好ましくないもので、経路枝数制限は重要な制約条件となる。流量配分則も、交通流等では、各車が管理者の指示通りに動くとは限らないので、考慮すべき条件である。本研究では流量配分則として均等配分則を取り上げたが、これは最も単純な流量配分則であるという意味もあるが、応用上も、前述の通信網の応用だけで無く、分岐点での経路選択が無意識に行なわれるもの（水道管内の水の流れ等）や、経路選択の有効な情報が全く無いもの（情報装置が故障した状態の道路網や巨大迷路等）、がこの規則に従うと考えられる。

第4章で提案した積上げ法は、ダイナミックルーティング方式における迂回候補群作成問題に対してのものだが、ダイナミックルーティング網の設計や評価にも使用できることが分かってきた[MATU 93] [INOUE 94]。

本研究の成果は、こういった様々な問題に対しても貢献できると考えられる。

実際の通信網上の問題には様々な制約があり、それら全てを考慮して定式化し最適化するのとは不可能である。しかし、本論文で扱った様に、一般性のある制約に着目して分析していく事で、今後も新しい有効なアルゴリズムが発見され、多くの問題が解決されていくであろう。

文 献

- [AHUJ 89] Ahuja R.K., Magnanti T.L. and Orlin J.B., "Network Flows," in Nemhauser G.L., Rinnooy Kan A.H.G. and Todd M.J. Edit. "Handbooks in Operations Reserch and Management Science vol. 1: Optimization," Chapter IV, North-Holland, Amsterdam (1989).
- [AKIY 73] 秋山稔, "近代通信交換工学," 電気書院 (1973).
- [ASH 81] Ash G.R., Cardwell R.H., and Murray R.P., "Design and Optimization of Networks with Dynamic Routing," Bell Syst. Tech. J., Vol.60, No.8, pp.1787-1820 (1981).
- [ASH 85] Ash G.R., "Use of a Trunk Status Map for Real Time DNHR," ITC11, session 4.4A, paper 4, Kyoto (1985).
- [ASH 90] Ash G.R., "Design and Control of Network with Dynamic Nonhierarchical Routing," IEEE Commun. Mag., Vol.28, No.10, pp.34-41 (1990).
- [ASH 91] Ash G.R., Chen J.-S., Frey A.E. and Huang B.D., "Real-Time Network Routing in a Dynamic Class-of-Service Network," ITC13, pp.187-194 (1991).
- [AWER 93a] Awerbuch B. and Leighton T., "A Simple Local-Control Approximation Algorithm for Multicommodity Flow," FOCS, pp.459-468 (1993).
- [AWER 93b] Awerbuch B. and Leighton T., "Multicommodity Flows: A Survey of Recent Research," ISSAC'93 Proceedings, LNCS #762, pp. 297-302 (1993).
- [BEIN 67] Beineke L. W. and Harary F., "The Connectivity Function of a Graph," Mathematika, No. 14, pp. 197-202 (1967).
- [BERG 73] Berge C., "Graphs and Hypergraphs," North Holland, Amsterdam (1973).
- [BOLL 78] Bollobas B., "Extremal Graph Theory," Academic Press, London (1978).
- [CHAU 91] Chaudhary V.P., Krishnan C.D. and Pack C.D., "Implementing Dynamic Routing in the Local Telephone Companies of USA," ITC13, pp.87-91 (1991).

- [CHEM 90] Chemouil P., Filipiak J. and Gauthier P., "Performance Issues in the Design of Dynamically Controlled Circuit-Switched Networks," IEEE Commun. Mag., Vol.28, No.10, pp.90-95 (1990).
- [COOK 71] Cook S.A., "The Complexity of Theorem-Proving Procedures," Proc. 3rd Ann. ACM Symp. on Theory of Computing, Association for Computing Machinery, New York, pp. 151-158 (1971).
- [DANT 63] Dantzig G.B., "Linear Programming and Extensions," Princeton University Press (1963). (邦訳：小山昭雄訳, "線形計画法とその周辺," ホルト・サウンダース (1983).)
- [DEBR 46] de Bruijn N.G., "A Combinatorial Problem," in Proc. Akademie Van Westeschappen, Vol. 49, part 2, pp. 758-764 (1946).
- [DINI 78] Dinits E.A., "Algorithms for solution of a problem of maximum flow in a network with power estimation", Soviet Math. Dokl., Vol. 11, pp. 1277-1280 (1970).
- [ELSA 88] Elsayed H.M., Mahmoud M.S., Bilal A.Y., and Bernussou J., "Adaptive Alternate-Routing in Telephone Network: Optimal and Equilibrium Solutions," Informations and Decision Technics, Vol. 14, No. 1, pp. 65-74 (1988).
- [ESWA 76] Eswaran K.P. and Tarjan R.E., "Augmentation Problem," SIAM J. Comput., Vol. 5, No. 4, pp. 653-665 (1976).
- [EVEN 75] Even S. and Tarjan R.E., "Network Flow and Testing Graph Connectivity," SIAM J. Comput. Vol. 4, NO. 4, pp. 507-518 (1975).
- [EVEN 76] Even S., Itai A. and Shamir A., "On the Complexity of Timetable and Multicommodity Flow Problems", SIAM J. Comput., 5, 4, pp. 691-703 (1976).
- [FORD 56] Ford L.R., Jr. and Fulkerson D.R., "Maximal Flow Through a Network," Canad. J. Math., Vol. 8, pp. 399-404 (1956).
- [FRAN 70] Frank H. and Chou W., "Connectivity Considerations in Design of Survivable Networks," IEE Transactions on Circuit Theory, Vol. CT-17, No. 4, pp. 486-490 (1970).
- [FRAN 90] Frank A., "Augmenting Graphs to Meet Edge-Connectivity Requirements," Proc. 31st IEEE FOCS, pp. 708-718 (1990).
- [FRAN 93] Frank A., Nagamochi H. and Ibaraki H., "On Sparse Subgraphs

- Preserving Connectivity Properties," J. of Graph Theory, Vol. 17, No. 3, pp. 275-281 (1993).
- [FRED 81] Frederickson G.N. and Ja'ja J., "Approximation Algorithms for Several Graph Augmentation Problems," SIAM J. Comput., Vol. 10, No. 2, pp. 270-283 (1981).
- [FUKU 84] Fukushima M., "A Nonsmooth Optimization Approach to Nonlinear Multicommodity Network Flow Problems," J. of the Operations Reserch Society of Japan, Vol. 27, No. 2, pp. 151-177 (1984).
- [GARE 79] Garey M.R. and Johnson D.S.: "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, San Francisco (1979).
- [GARG 93] Garg N., Santosh V. S. and Singla A.: "Improved Approximation Algorithms for Biconnected Subgraphs via Better Lower Bounding Techniques," Proc. 4th ACM-SIAM Symp. on Discrete Algorithms, pp. 103-111 (1993).
- [GIMP 74] Gimpelson L.A., "Network Management: Design and Control of Communications Networks," Electorical Communication, 49,1 (1974).
- [GIRA 90] Girard A., "Routing and Dimensioning in Circuit-Switched Networks," Addison-Wesley Publishing Company, Massachusetts, (1990).
- [HAN 92] Han X., Kelson P., Ramachandran V. and Tarjan R. E.: "Computing Minimal Spanning Subgraphs in Linear Time," Proc. 3rd ACM-SIAM Symp. on Discrete Algorithms, pp. 146-156 (1992).
- [HARA 69] Harary F., "Graph Theory," Addison-Wesley Publishing Company, Massachusetts (1969). (邦訳：フランク・ハラリイ著, 池田貞雄訳, "グラフ理論," 共立出版 (1971).)
- [HASI 86a] 橋本, 中村, "I N S に向けた電気通信網基本計画," N T T 施設, Vol.38, No.8, pp.10-16 (1986).
- [HASI 86b] 橋本, 岡野, "ネットワークの効率的建設を目指して," N T T 施設, Vol.38, No.8, pp.32-39 (1986).
- [HAYA 92] 林, 大槻, "ネットワークの2分割によるエンド・エンド経路分散の検討," 電子情報通信学会秋季大会講演論文集, 分冊1, pp. 2 (1992).

- [HURL 87] Hurley B.R., Seidl C.J.R., and Sewell W.F., "A Survey of Dynamic Routing Methods for Circuit-Switched Traffic," IEEE Commun. Mag., Vol.25, No.9, pp.13-21 (1987).
- [IBAR 83] 茨木俊秀, "講座・数理計画法 8 組合わせ最適化 分枝限定法を中心として," 産業図書 (1983).
- [IBAR 89a] 茨木俊秀, "グラフのdisjoint pathについて," 第1回RAMPシンポジウム論文集, pp. 3-14 (1989).
- [IBAR 89b] 茨木俊秀, "アルゴリズムとデータ構造," 昭晃堂 (1989).
- [IBAR 93] 茨木俊秀, 福島雅夫, "情報数学講座 1 4 最適化の手法," 共立出版 (1993).
- [INOUE 89] Inoue A., Yamamoto H., and Harada Y., "An Advanced Large-Scale Simulation System for Telecommunication Networks with Dynamic Routing," Networks89, pp.77-82 (1989).
- [INOUE 91] Inoue A., Mase K., Yamamoto H., and Suyama M., "A State- and Time-Dependent Routing Scheme for Telephone Networks," ITC13, pp.195-200 (1991).
- [INOUE 92] 井上明也, 伊藤大雄, "通信網管理への数理計画法の適用について," 第4回RAMPシンポジウム論文集, pp. 57-65 (1992).
- [INOUE 94] Inoue A., Ito H. and Satake T., "Dynamic-Routing Network Design under Transitional Conditions," Networks'94, P.6.1 (1994).
- [IRI 93] Iri M., "Essay in the Theory of Uncontrollable Flow and Congestion," RAMP Reserch Meeting, Nov. 20 (1993).
- [ITO 91] 伊藤大雄, 井上明也, "STRにおける迂回候補群作成法の予測誤り耐力の評価," 電子情報通信学会秋季大会講演論文集, 分冊3, pp. 131, (1991).
- [ITO 92a] 伊藤大雄, "流量割当制約付き多品種流問題の計算量," 電子情報通信学会技術研究報告, No. 91, Vol. 449, COMP91-87, pp.27-33 (1992).
- [ITO 92b] 伊藤大雄, "経路枝数に制限を持つ多品種流問題," 電子情報通信学会論文誌A, Vol. J75-A, No. 3, pp.643-645 (1992).
- [ITO 92c] 伊藤大雄, 井上明也, "ダイナミックルーティングのための迂回候補群作成法," 電子情報通信学会論文誌B-I, Vol. J75-B-I, No. 5, pp. 323 -332 (1992).

- [ITO 92d] 伊藤大雄, 山田慈朗, 佐竹康宏, 井上明也, "ダイナミックルーチング(STR)の迂回ルート候補設定機能," NTT R&D, Vol. 41, No. 6, pp.697-704 (1992).
- [ITO 92e] 伊藤大雄, 井上明也, "ダイナミックルーチングSTRのための呼量予測アルゴリズムの評価," 電子情報通信学会秋季大会講演論文集, 信学会, 分冊3, pp. 72 (1992).
- [ITO 94a] 伊藤大雄, "領域を考慮したグラフの連結性問題," 第7回回路とシステム軽井沢WS論文集, pp. 321-326 (1994).
- [ITO 94b] 伊藤大雄, "グラフにおける節点・領域連結度について," 電気学会論分誌C, Vol.114-C, No.4 (1994).
- [ITO 94c] 伊藤大雄, 永持仁, "T-混合カットにおける領域間連結度の性質," OR春, 5 (1994).
- [ITO 94d] 伊藤大雄, "均等配分する多品種流問題の計算量," 電子情報通信学会論文誌A, Vol. J77-A, No. 11 (1994). (掲載予定)
- [ITO 94e] Ito H., "Node-to-Area Connectivity of Graphs," Proc. APORS'94 (1994). (掲載予定)
- [ITO 95] Ito H., "Connectivity problem on area graphs for locally striking disasters — Direct NA-connection," IEICE Transactions, Special Section of Selected Papers from the 7th Karuizawa Workshop on Circuits and Systems (1995). (掲載予定)
- [IWAN 93] 岩野和生, "ネットワークフロー問題の最近の進展," 藤重悟編, "離散構造とアルゴリズムII," 第3章, 近代科学社 (1993).
- [JAGE 74] Jagerman D.L. "Some Properties of the Erlang Loss Function," Bell system Technical J., Vol. 53, No. 3, pp. 525-551 (1974).
- [KARM 84] Karmarkar N., "A New Polynomial-time Algorithm for Linear Programming," Combinatorica, Vol. 4, pp. 373-395 (1984).
- [KARP 72] Karp R.M., "Reducibility among Combinatorial Problems" in Complexity of Computer Computations, ed. Miller R.E. and Thatcher J.W., pp. 85-104, Plenum Press, New York (1972).
- [KAUT 68] Kautz W.H., "Bounds on Directed (d,k)-graphs," in Theory of Cellular Logic Networks and Machines, AFCRL-68-0668, SRI proj. 7258, Final Report (1968).
- [KELS 91] Kelson P. and Ramachandran V. "On Finding Minimal 2-Connected

- Subgraphs," Proc. of 2nd ACM-SIAM Symp. on Discrete Algorithms, pp. 178-187 (1991).
- [KENN 80] Kennington J.L. and Helgason R.V., "Algorithms for Network Programming," A Wiley and Sons, New York, (1980).
- [KEY 90] Key P.B. and Cope G.A., "Distributed Dynamic Routing Schemes," IEEE Commun. Mag., Vol.28, No.10, pp.54-64 (1990).
- [KHAC 79] Khachiyan L.G., "A Polynomial Algorithm in Linear Programming," Dokl. Akad. Nauk SSSR N.S. 244:5, pp. 1093-1096 (1979). [English transl., Soviet Math. Dokl. 20:1, pp. 191-194 (1979).]
- [KHUL 92] Khuller S. and Vishkin U.: "Biconnectivity Approximations and Graph Carvings," Proc. 24th ACM Symp. Theory of Computing, pp. 759-770 (1992).
- [KING 92] King V., Rao S. and Tarjan R.E., "A faster deterministic maximum flow algorithm", 3rd ACM-SIAM Symp. on Discrete Algorithms, pp. 157-164 (1992).
- [KRUP 82] Krupp R.S., "Stabilization of Alternate Routing Network," ICC-82, pp.31.2.1-31.2.5 (1982).
- [LADN 75] Ladner R.E., "On the Structure of Polynomial Time Reducibility," J. of the Assoc. for Comput. Machinery, Vol. 22, No. 1, pp. 155-171 (1975).
- [LAWL 76] Lawler E.L., "Combinatorial Optimization: Networks and Matroids," Holt, Rinehart and Winston, New York (1976).
- [MASE 90] Mase K., Yamamoto H., "Advanced Traffic Control Method for Network Management," IEEE Commun. Mag., Vol.28, No.10, pp.82-88 (1990).
- [MATU 93] 松崎隆一, 伊藤大雄, "フロー割当によるネットワーク性能推定法," 電子情報通信学会技術研究報告, Vol.93, No.259, IN93-93, pp. 135-140 (1993).
- [MENG 27] Menger K.: "Zur Allgemeinen Kurventheorie," Fund. Math. No. 10, pp. 96-115 (1927).
- [NAGA 88] Nagamochi H. "Studies on Multicommodity Flows in Directed Networks," Doctoral Dissertation of Engineering, Applied Mathematics and Physics, Kyoto Univ. (1988)

- [NAGA 89] Nagamochi H. and Ibaraki T., "An Efficient Algorithm for the Multicommodity Flow Problem with Many Commodities," Technical Report, 89008, Kyoto Univ. (1989).
- [NAGA 92] Nagamochi H. and Ibaraki T., "A Linear-Time Algorithm for Finding a Sparse k -Connected Spanning Subgraph of a k -Connected Graph," *Algorithmica*, No. 7, pp. 583-596 (1992).
- [NAGA 93] 永持仁, "グラフの最小カットについて," 藤重悟編, "離散構造とアルゴリズムII," 第4章, 近代科学社 (1993).
- [NAOR 90] Naor D., Gusfield D. and Martel C., "A Fast Algorithm for Increasing the Edge-Connectivity," *Proc. 31st Annual IEEE Symp. Found. Comp. Sci*, St Louis, pp. 698-707 (1990).
- [ONO 91] 小野正, "経路リンク数を考慮した多品種フロー問題," 特別研究報告書, 京都大学工学部数理工学科, 3 (1991).
- [PAPA 83] Papadimitriou C.H. and Steiglitz K., "Combinatorial Optimization: Algorithms and Complexity," Prentice-Hall, Inc. (1983).
- [PERL 78] Perl Y. and Shiloach Y., "Finding Two Disjoint Paths between Two Pairs of Vertices in a Graph", *J. Assoc. Comput. Mach.*, Vol. 25, pp. 1-9 (1978).
- [REGN 90] Regnier J., Cameron W.H., "State-Dependent Dynamic Traffic Management for Telephone Networks," *IEEE Commun. Mag.*, Vol.28, No.10, pp.42-53 (1990).
- [SATA 93] 佐竹孝, 伊藤大雄, "節点・領域間経路の性質と袋小路の解消法," *OR学会秋季研究発表会論文集*, pp. 46-47 (1993).
- [SONE 88] 曾根岡昭直, "高信頼網構成に関するグラフ理論的研究," 博士論文, 東京大学工学部 (1988).
- [TARD 85] Tardos E., "A Strongly Polynomial Minimum Cost Circulation Algorithm," *Combinatorica*, Vol. 5, No. 3, pp. 247-255 (1985).
- [WATA 92] Watanabe T., Mashima T. AND Taoka S., "The k -Edge-Connectivity Augmentation Problem of Weighted Graphs," *3rd ISAAC'92, LNCS #650*, pp. 31-42 (1992).
- [WATA 93] Watanabe T. and Nakamura A., "A Minimum 3-Connectivity Augmentation of Graph," *J. of Computer and System Sciences*, 46, pp. 91-128 (1993).

付録 1 補題3.1の証明

補題3.1を証明する前に、最大流問題MF（1. 3 参照）における、次の性質を示しておく。

【補題A.1】 MFに最適解（最大流）が存在するならば、単純流に分解できる最適解が存在する。□

本補題は、最大流問題が増大路法(augment path method)[FORD 59] [IWAN 93]で解けることを用いて証明する。増大路法の概要を以下に示す。使用する記号は1. 3のMFの定義に従う。

初期値として流れを $f(a)=0, a \in A$ とし、次の更新操作に従って流れ $f(a)$ を更新していく。

更新操作

$G=(V,A)$, 容量 $u(a), a \in A$, 現時点で得られている流れ $f(a), a \in A$ に基づいて、残余ネットワーク $N_f=(G_f=(V,A_f), u_f)$ （但し $u_f: A_f \rightarrow \{\text{正実数}\}$ は枝の容量）を以下のように構成する。

$$A_f = A_f^1 \cup A_f^2$$

$$A_f^1 = \{a \in A \mid f(a) < u(a)\}$$

$$A_f^2 = \{a(x,y) \mid a(y,x) \in A, f(a(y,x)) > 0\}$$

$$u_f(a) = u(a) - f(a), a \in A_f^1$$

$$u_f(a(x,y)) = f(a(y,x)), a(x,y) \in A_f^2$$

G_f 上の s, t 間の有向路 p （これを増大路と呼ぶ）を選び、 p 上の枝の容量 u_f の最小値を $v(p)$ （これを更新量と呼ぶ）とする。そして $f(a)$ を次の様に更新する。

$$f(a) = f(a) + v(p), a \in A_f^1 \cap A(p)$$

$$f(a(x,y)) = f(a(x,y)) - v(p), a(y,x) \in A_f^2 \cap A(p)$$

以上の更新操作を繰り返し、増大路が発見できなくなった時点で得られている流れ $f(a), a \in A$ が最大流を与える。

増大路法は増大路の見つけ方の違いで種々のアルゴリズムが提案されている。しかし、最短路を見つけて増大路とする方法によって、この繰り返しが有限回（実は多項式回）で終了することが分かっている [FORD 59] [IWAN 93]。このことは、直感的に言えば、MFの最適解は、増大路に添った流れから構成されている、と考えられる。但し残余ネットワークでは逆向きの枝 (A_f^2) も考えるので、増大路は G 上の有向路となるとは限らず、増大路の集合をそのまま単純流分解と考えることはできない。しかし、増大路の集合を元に簡単に単純流分解を求めることができる。そのことを、次に示し、補題A.1の証明とする。

補題A.1の証明) 増大路法によって得られた最適解を $f^*(a)$, $a \in A$ 、そこで得られた増大路の集合を $Q = \{p_1, p_2, \dots, p_k\}$ とする。 p_i の際の更新量を $v(p_i)$ とする。 p_i , $i=1, 2, \dots, k$ は G 上の有向路であるとは限らないが、 G の方向を無視して無向グラフと考えたならば、路になる。よってこれらは無向路と仮に呼ぶことにする。また、有向路でない無向路を真の無向路と呼ぶことにする。（但しこれらの呼称はこの付録1内でのみ用いる。） p_i , $i=1, 2, \dots, k$ が全て G 上の有向路であったならば、 p_i を通る流量 $v(p_i)$ の単純流 f_i を考えれば、明かに f_i , $i=1, 2, \dots, k$ が f^* の単純流分解になっている。よって真の無向路が存在する場合を考える。

逆向きに使用されている枝 $a(x, y) \in A$ を一つ選ぶ。 $a(x, y)$ を逆向きに使用している真の無向路 p_i , $i \in \{1, 2, \dots, k\}$ を一つ取り出す。増大路のアルゴリズムより、 $a(x, y)$ を正しい向きに通る路 p_j , $j \in \{1, 2, \dots, k\} - \{i\}$ が存在する（図A.1(a)参照）。 p_i の s から y までの部分路を $p_i(y)$ 、 x から t までの部分路を $q_i(x)$ 、 p_j の s から x までの部分路を $p_j(x)$ 、 y から t までの部分路を $q_j(y)$ 、と表すとすると、 p_i と p_j は

$$p_i = \langle p_i(y), a(x, y), q_i(x) \rangle$$

$$p_j = \langle p_j(x), a(x, y), q_j(y) \rangle$$

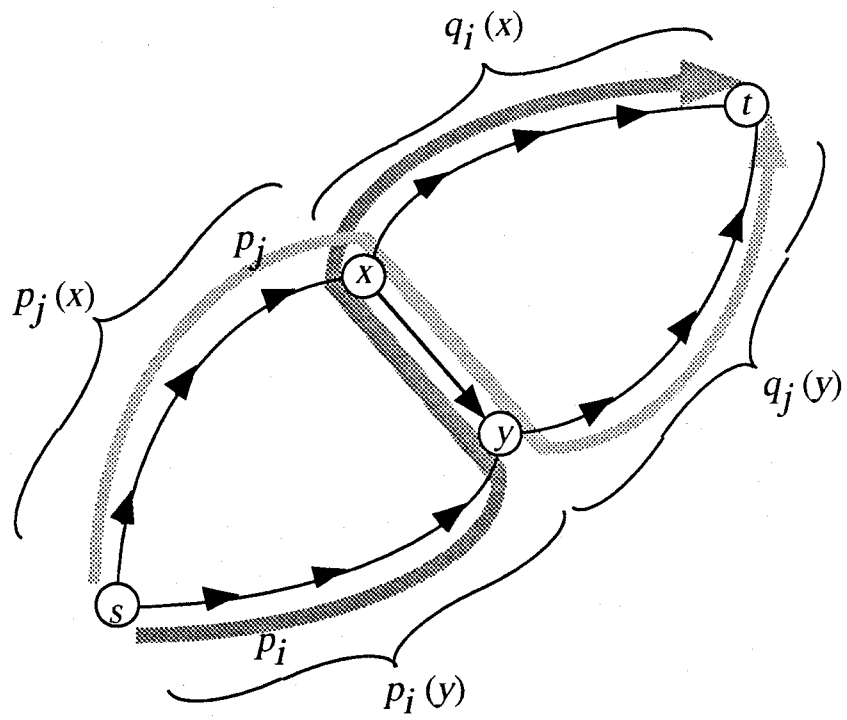
の様に表現できるここで、 $v(p_i) \geq v(p_j)$ と仮定する（※）。そこで、

$$p_i' = \langle p_i(y), q_j(y) \rangle$$

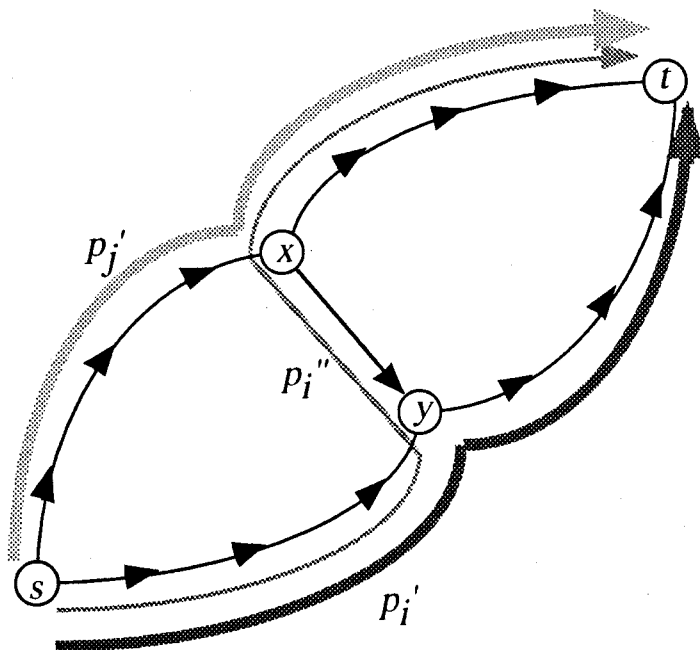
$$p_j' = \langle p_j(x), q_i(x) \rangle$$

$$p_i'' = p_i$$

なる新たな3つの無向路を考え（図A.1(b)参照）、



(a) p_i and p_j



(b) p_i' , p_j' and p_i''

図A.1 補題A.1の証明の説明

Fig. A.1. Proof of Lemma A.1.

$$v(p_i') = v(p_i)$$

$$v(p_j') = v(p_j)$$

$$v(p_i'') = v(p_i) - v(p_j)$$

とする。すると

$$Q' = Q - \{p_i, p_j\} \cup \{p_i', p_j', p_i''\}$$

はやはり f^* を得る増大路の集合となり、しかも枝を逆向きに流れる量が減っている（この場合 $v(p_j)$ だけ減っている）。しかも $a(x, y)$ を逆向きあるいは正しい向きに選ぶ増大路の数の合計が必ず 1 つは減少する。※の部分で $v(p_i) \leq v(p_j)$ と仮定した場合も同様の議論が成立する。よって、有限回の繰り返して、 $a(x, y)$ を逆向きに選ぶ増大路が無くすることが出来る。逆向きに使用されている $a(x, y)$ に順にこの操作を適用することで、最終的に全ての無向路を有向路とすることが出来る。全て有向路に出来たならば、前述の様に、単純流分割を得る。 Q.E.D.

補題A.1を使って補題3.1を証明することが出来る。

補題3.1の証明) CMCfの問題例 $N=(G=(V, A), K, S, d, u, c)$ の最適解を f とする。 f を元に、品種 $k \in K$ に対応して、有向グラフ $G=(V, A)$ 上の容量関数 u_k を

$$u_k(a) = f(a, k), a \in A$$

とする。MFの問題例 (G, u_k) は

$$f_k(a) = f(a, k), a \in A$$

を実行可能解として持ち、しかも全ての容量が有限値なので、最適解（最大流）をもつ。よって補題3.0.1から単純流に分解できる最適解 g_k が存在する。 g_k の単純流の分解を $g_k^1, g_k^2, \dots, g_k^H$ とし、それぞれの流量を $v_k^1, v_k^2, \dots, v_k^H$ とする。最適解 g_k の流量 v_k は実行可能解の流量 $d(k)$ 以上である、すなわち、

$$v_k = \sum_{i=1}^H v_k^i \geq d(k)$$

が成り立つ。すると、ある整数 $h \in \{2, 3, \dots, H\}$ が存在して

$$\sum_{i=1}^{h-1} v_k^i < d(k) \quad \text{かつ} \quad \sum_{i=1}^h v_k^i \geq d(k)$$

を満たす。 g_k^h と通る路が等しく、流量が $v_k^{h'} = d(k) - \sum_{i=1}^{h-1} v_k^i$ である単純流を $g_k^{h'}$ とする。すなわち、 $g_k^1, g_k^2, \dots, g_k^{h-1}, g_k^{h'}$ の和で表される流れ g_k' は、 $s(k), t(k)$ 間の流量 $d(k)$ の流れとなり、明かに流量保存則を満足する。各品種 $k \in K$ について、上記の操作で流れ g_k' を求め、それらを合わせて N 上の流れ g' を作る、すなわち

$$g'(a,k)=g'_k(a), a \in A, k \in K$$

とする。

$$g'(a,k)=g'_k(a) \leq g_k(a) \leq u_k(a)=f(a,k), a \in A, k \in K \quad (1)$$

なので、 g' は容量制約を満たす。また、各 g'_k が流量保存則を満足することから、明かに g' は流量保存則を満足する。以上から、 g' は N の実行可能流である。しかも、(1)より、 g' の費用は f の費用以下であるので、 g' は N の最適解である。 g' はその構成法より、単純流に分解できる。 Q.E.D.

付録 2 . 定理3.4の証明

既知のNP完全問題である充足可能性問題SATを帰着する。SATの定義は2. 2. 1 参照。

SATの任意の問題例 (U, C) , $(U=\{u_1, u_2, \dots, u_N\}, C=\{C_1, C_2, \dots, C_M\})$ に基づいて、品種数が2であるMCRの問題例を構成し、両問題の解 (yesかnoか) が一致する様にする。

節点集合

$$V = \{u_i^s, u_i^t \mid i=1, 2, \dots, N\} \cup \{u_i^j, \neg u_i^j \mid i=1, 2, \dots, N, j=1, 2, \dots, 2p_i\} \\ \cup \{\neg u_i^j \mid i=1, 2, \dots, N, j=1, 2, \dots, 2q_i\} \\ \cup \{C_k \mid k=1, 2, \dots, M\} \cup \{s_1, t_1, s_2, t_2\}$$

(但し、 p_i は u_i が含まれる節 (C_j) の総数、 q_i は $\neg u_i$ が含まれる節の総数。)

枝集合

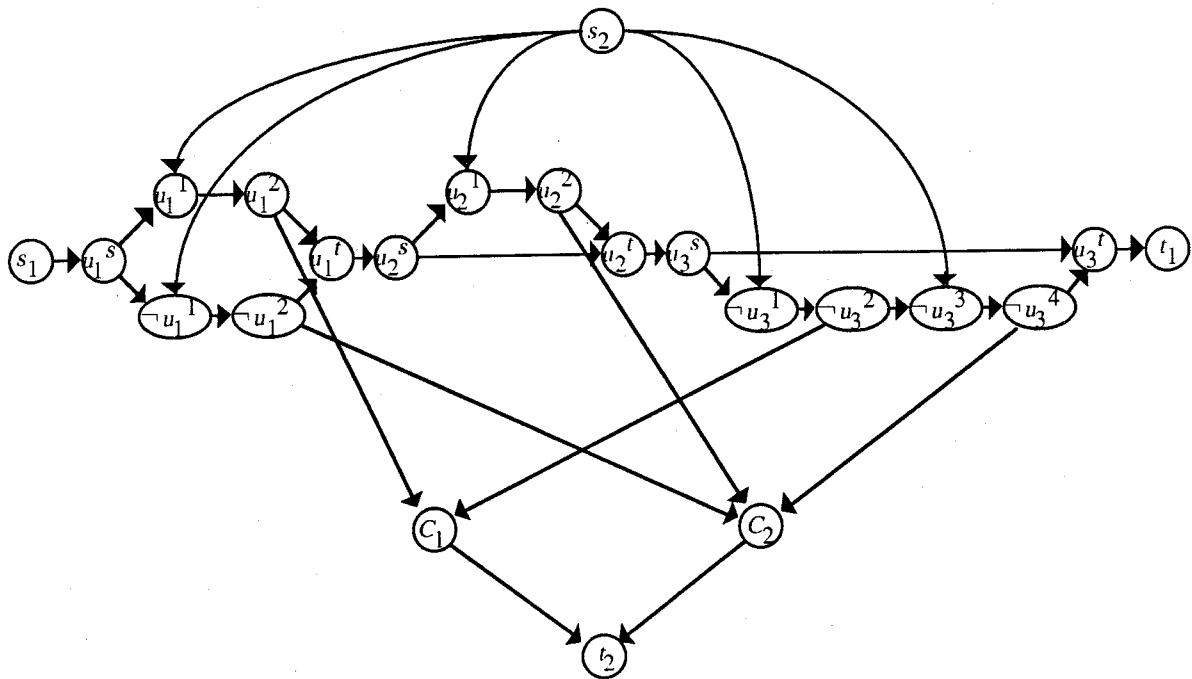
$$E = \{a(u_i^j, u_i^{j+1}) \mid j=1, 2, \dots, 2p_i-1\} \cup \{a(\neg u_i^j, \neg u_i^{j+1}) \mid j=1, 2, \dots, 2q_i-1\} \\ \cup \{a(u_i^t, u_{i+1}^s) \mid i=1, 2, \dots, N-1\} \\ \cup \{a(s_2, u_i^j), a(s_2, \neg u_i^j) \mid i=1, 2, \dots, N-1, j=1, 2, \dots, 2p_i, j \text{は奇数}\} \\ \cup \{a(s_1, u_1^s), a(s_1, \neg u_1^s), a(u_N^t, t_1), a(\neg u_N^t, t_1)\} \\ \cup \{a(u_i^{2h}, C_k) \mid C_k \text{は} u_i \text{を含む} h \text{番目の節}\} \\ \cup \{a(\neg u_i^{2h}, C_k) \mid C_k \text{は} \neg u_i \text{を含む} h \text{番目の節}\} \\ \cup \{a(C_k, t_2) \mid k=1, 2, \dots, M\}$$

枝の容量は全て1とする。

品種1の要求流量は1、品種2の要求流量は M 。

SATの問題例 $U=\{u_1, u_2, u_3\}$, $C_1=\{u_1, \neg u_3\}$, $C_2=\{\neg u_1, u_2, \neg u_3\}$ に基づいて構成されたネットワークを図A.2に示す。

両問題の解が一致することは容易に確かめられるので、本稿では省略する。詳細は原論文[EVEN 76]参照。 Q. E. D.



図A.2 充足可能性問題から、2 品種MCRへの帰着例

Fig. A.2 An example of reduction from SAT to MCW with 2-commdities.

付録 3. 回線留保

本付録では、回線留保方式の概略を説明する。詳細は[AKIY 73] [KRUP 82]を参照。

迂回を行なうと、通常 1 本の枝で疎通する呼を 2 本の枝を用いて疎通することになる。つまり、迂回呼は基本呼（基本路を通る呼）に比べて、設備使用能率（疎通呼量／使用設備数）が半分になる。よって、迂回は、回線に余裕がある場合に行なうべきであり、基本呼を押し退けてまで迂回呼を疎通すると、かえって設備の使用効率を悪化させることになる。基本呼を圧迫せずに適切な量を迂回させる為に用いられる措置が、回線留保である。

具体的には、回線にあらかじめ設定した数（留保数）より多い空き回線が存在するときのみ、迂回呼（非優先呼）の接続を許すという制御である。これにより、回線が混んでいる時には、迂回呼は接続されなくなり、基本呼（優先呼）が圧迫されることが少なくなる。適切な留保数は回線の太さに依存して変化するが、その変化は小さく、通常数本程度で十分とされる。

回線留保を行なった場合は、基本呼と迂回呼の呼損率は当然異なる。回線数 n 、留保回線数 m 、優先呼（基本呼）量 trf_1 、非優先呼（迂回呼）量 trf_2 の時の優先呼と非優先呼の呼損率 B_1, B_2 はそれぞれ、以下の式で算出される。

$$B_1=p_n, B_2=\sum_{r=n-m}^n p_r$$

但し

$$p_r=\begin{cases} \frac{(trf_1+trf_2)^r}{r!}p_0, & 0\leq r\leq n-m \\ \frac{(trf_1+trf_2)^{n-m}(trf_1)^{r-n+m}}{r!}p_0, & n-m+1\leq r\leq n \end{cases}$$

$$p_0=\left(\sum_{r=0}^{n-m} \frac{(trf_1+trf_2)^r}{r!} + \sum_{r=n-m+1}^n \frac{(trf_1+trf_2)^{n-m}(trf_1)^{r-n+m}}{r!}\right)^{-1}$$

である。

付録4．繰り返し法のアゴリズム

本付録には繰り返し法のアゴリズムITERATE-LPを掲げる。

[記号の定義]

b_0 : 呼損率基準値 (通常0.01)

$trf_{UL}(v,w)$: $trf_U(v,w)$ の下限

$trf_{LL}(v,w)$: $trf_L(v,w)$ の下限

その他の記号の定義は本文4．3．1と同じ。

procedure ITERATE-LP

begin

1 begin comment 初期値の設定;

2 $trf_{UL}(v,w) := trf_0$, 但し $b_0 = B_2(n(v,w), m(v,w), trf_1(v,w), trf_0)$

3 for each $a(v,w) \in A$ do

4 $trf_{LL}(v,w) := 0$

5 for each $u \in V, u \neq v, w$ do

6 $x(v,u,w) := (|V|-2)^{-1}$

end;

7 step:=1

end;

8 begin

9 次の連立方程式を解き $\alpha(v,w), trf_2(v,w), b_1(v,w), b_2(v,w)$ を求める。

$$\alpha(v,w) = trf_1(v,w)b_1(v,w), \quad \forall a(v,w) \in A$$

$$trf_2(v,w) = \sum_{u \in V} x(v,w,u)\alpha(v,u)(1-b_2(w,u))$$

$$+ \sum_{u \in V} x(u,v,w)\alpha(u,w)(1-b_2(u,v))$$

$$b_1(v,w) = B_1(n(v,w), m(v,w), trf_1(v,w), trf_2(v,w)), \quad \forall a(v,w) \in A$$

$$b_2(v,w) = B_2(n(v,w), m(v,w), trf_1(v,w), trf_2(v,w)), \quad \forall a(v,w) \in A$$

end;

10 if step>1 and $trf_2(v,w) \geq trf'_2(v,w)$ then $trf_{LL}(v,w) := trf_L(v,w)$;

11 $trf_L(v,w) := (trf_{LL}(v,w) + trf_2(v,w))/2$

12 $trf_U(v,w) := \max(2trf_2(v,w), trf_{UL}(v,w))$

```

13   $x'(v,u,w) := x(v,u,w)$ 
14  begin
15    次のLPを解き  $x(v,w,u)$ を求める
      [目的関数]  $\sum_{a(v,w) \in A} trf_c(v,w) \rightarrow \max$ 
      [制約条件]
           $trf_c(v,w)$ 
           $= trf_1(v,w)(1-b_1(v,w)) + \sum_{u \in V} x(v,u,w) o(v,w)(1-b_2(v,u))(1-b_2(u,w)),$ 
 $a(v,w) \in A$ 

           $trf_L(v,w)$ 
           $\leq \sum_{u \in V} x(v,w,u) o(v,u)(1-b_2(w,u)) + \sum_{u \in V} x(u,v,w) o(u,w)(1-b_2(u,v))$ 
           $\leq trf_L(v,w), a(v,w) \in A$ 
           $\sum_{u \in V} x(v,u,w) \leq 1, a(v,w) \in A$ 
           $x(v,u,w) \geq 0, a(v,w) \in A, u \in V$ 

      end;
16  begin comment 収束判定;
17    if  $|x(v,u,w) - x'(v,u,w)| < \varepsilon$ , for all  $a(v,w) \in A, u \in V, u \neq v, w$  then stop;
      end;
18   $trf_2'(v,w) := trf_2(v,w)$ 
19  step := step + 1
20  goto line 8
end.

```